9-13-2012

# Quantifying Performance Bias in Label Fusion

Alexander M. Venzin

**QUANTIFYING PERFORMANCE BIAS IN LABEL FUSION**

THESIS


Alexander M. Venzin, B.A.



AFIT/GAM/ENC/12-04



**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

# AIR FORCE INSTITUTE OF TECHNOLOGY

**Wright-Patterson Air Force Base, Ohio**

AFIT/GAM/ENC/12-04

# QUANTIFYING PERFORMANCE BIAS IN LABEL FUSION

THESIS

Presented to the Faculty

Department of Mathematics and Statistics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Applied Mathematics

Alexander M. Venzin, B.A.

September 2012

AFIT/GAM/ENC/12-04

QUANTIFYING PERFORMANCE BIAS IN LABEL FUSION

Alexander M. Venzin B.A.

Approved:

| | |
|---|---|
| Dr. Christine Schubert-Kabban (Chairman) | date |
| Dr. Mark E. Oxley (Member) | date |
| Dr. James D. Cordeiro (Member) | date |

*Table of Contents*

Page

Page

Page

Page

*List of Figures*

Figure Page

Figure Page

Figure Page

Figure Page

*List of Tables*

AFIT/GAM/ENC/12-04

*Abstract*

With respect to military applications, classification systems are employed to remotely assess whether an element of interest falls into a “target” class or “non-target” class. The name of the classes is arbitrary, but the role of the classification system remains the same. These systems also have uses in fields as far ranging as biostatistics to search engine keyword analysis. The performance of the system is often summarized as a trade-off between the proportions of elements correctly labeled as “target” plotted against the number of elements incorrectly labeled as “target.” The first term is generally regarded as the “hit rate” and the second term as the “false alarm rate.” These are empirical estimates of the true positive and false positive rates. These rates are often plotted to create a receiver operating characteristic (ROC) curve that acts as a visual tool to assess classification system performance. The performance of the system(s) can possibly be increased if the information provided by both systems can be fused together to create a new, combined system using any number of techniques. The research contained in this thesis focuses specifically on the label fusion technique and the bias that can occur when using incorrect assumptions regarding the partitioning of the event set. This partitioning may be defined in terms of what will be called *within* and *across* label fusion. The major goals of this thesis are the formulaic development and quantification of performance bias between different types of *across* and *within* label fusion and analysis of the effects of individual classification system performance, correlation, and target environment on the magnitude of bias between these two types of label fusion. Formulas developed may be used to adjust optimal parameters and performance measures to

appropriately reflect fused system performance on various platforms. Thus, this research can be applied in the future to address the inherent bias that may be built into fused classification systems.

# QUANTIFYING PERFORMANCE BIAS IN LABEL FUSION

## *I Introduction*

### *1.1 Background*

As technology advances and access to inexpensive and efficient computing resources continues to rise, there is reason to believe that sole reliance on the human element in any number of decision-based applications will be reduced and there will be an increased use of automated systems. Such technology has advantages. For example, if a machine can take the place of a human being in a potentially life threatening situation, why not rely on autonomous processes? However, as research into the area of *decision* and *classification system* theory has evolved, it has become clear that the process of human-based decision making is a difficult task for a machine. Most notably, there are a considerable number of ways to analyze data and make a decision.

Classically, the goal of *label fusion* is to combine the output from multiple classification systems to improve predictive accuracy. This makes intuitive sense from the point of view that different classification systems excel at classifying different events. Further, the performance of different combinations of classification systems depends in large part on the fusion rule to be used. There are times, however, that fusing multiple classification systems together can actually decrease performance and in this instance, it may be best to not fuse and instead choose the optimally performing individual classification system.

Apart from fusion rule and choice of classification systems, the level of classification system dependence is a necessary consideration. Some methods make no assumption about the dependence of individual classification systems while others

inherently assume independence. For example, *probabilistic neural networks* make no assumptions regarding the level of dependence between individual classification systems (*Leap et al: 2008)*. Instead, the probabilistic neural network is tuned to the targets it is designed to classify by using a portion of the available data as training material. The theory that this thesis is based upon, label fusion, requires that the individual classification systems are independent in order to derive the functions for the receiver operating characteristic (ROC) curve. The ROC curve is a graphical tool for depicting true hit rate along the vertical axis (the number of target events correctly classified as targets, i.e., true positive rate) as compared to false alarm rate along the horizontal axis (the number of target events incorrectly classified as non-targets, i.e., false positive rate).

With the notions of fusion and classification system dependence in place, the focus becomes how to optimize performance given the systems and fusion rules available. It is common to apply a series of logical rules and combinations of logical rules and choose the combination that optimizes performance (*W. Khreich et al: 2012)*. Another common method is to treat the output of the individual classification systems as input to a neural network and run a regression-based analysis (*Leap et al: 2008)*. Along with optimization of performance comes the consideration of optimization of available resources. Fusing classification systems tuned to different target types, perhaps from legacy systems, invokes concerns regarding the probability with respect to target prevalence and event set partitions. However, systems combined using label fusion rules are especially prone to simplifying assumptions which may overlook target partitions. Mathematical and computational modeling generally performs well in this regard relative to diagnostic testing or actual engineering testing. As such, the fused system should

outperform any individual system and perform efficiently despite environmental constraints such as target prevalence and classification system correlation.

*1.2 Problem Statement*

The research contained herein addresses the quantification of bias between what will be defined as *across* and *within* label-fused classification systems and the effects of individual classification system performance, correlation, and target environment on such bias. Specifically, fused systems that do not account for differences in the partitioning of targets between fused systems produce errors, or a bias, in system performance. Depending on how the specific target partitions of the event set are defined for the classification systems to be fused gives rise to the notion of *within* versus *across* label fusion. It is in mistaking between what can be labeled as *within* and *across* fusion that the bias in system performance occurs. Thus, this research involves the derivation of formulas that allow for quick and easily implementable bias correction algorithms. Building on the work of (*Schubert: 2005), (Leap et al: 2008), (Schubert et al: 2005), (Oxley, Bauer: 2003)*, who have previously investigated and derived functions for the ROC curves of label-fused systems, the formulas for bias will be derived using these formulas. The algorithm developed here will create a graphical and analytical tool for measuring bias in the form of a *bias curve*. Using the work of (*Schubert: 2005)*, correlation between classification systems will be examined. Individual system performance and structure of the target environment is also considered.

*1.3 Research Methodology*

Prior to developing a functional form for the bias, some rudimentary classification system theory and label fusion theory must be derived along with the functions for ROC

curves under both Boolean *AND/OR* rules and *within* and *across* label fusion. The theory of correlation under these Boolean rules will be discussed briefly. With the ROC curves, the bias can be defined as the difference in true positive rate for two ROC curves under different label fusion rules. These formulas have been derived under the assumption of conditional independence with respect to the non-target partition of the event set. This will have important implications when it comes to assessing the correlation between different label-fused classification systems. A specific bias function is derived for comparing each type of *across* versus *within* label-fused system under a given Boolean rule. Using simulations, the performance of classification systems in different target environments under different assumptions of correlation and the bias between different combined label-fused systems are compared.

### *1.4 Assumptions*

The main assumptions made in this document have been largely inherited from previous research. First, it is assumed that there is a two-class label set. From the point of view of military application, the two-class label set may generally be composed of the target class and non-target class, though it need not be. Consider a simple example involving a two-class label set. Suppose a classification system is built to classify a specific type of radar signature for a type of enemy aircraft. This system could employ a vector of probability values that favors (that is, employs a higher weighting of probability) the classification of high velocity fighter jets. Perhaps other, less highly weighted parameters are concerned with classifying support aircraft. Now suppose that an array of these systems is put out into the field and a high velocity aircraft is correctly classified as a target. After a series of aircraft have been classified by the systems, the

performance of the array can be depicted graphically using the ROC curve where true positive or hit rate is plotted along the vertical axis and the false positive or false alarm rate is plotted along the horizontal axis.

Notice that the classification systems themselves did not identify the actual aircraft, but rather used classification systems to classify the aircraft into a specific class, e.g., a target. This is a subtlety that is inherent with classification system fusion: the classification systems cannot identify the elements of interest, only assign a class label determined by pre-defined thresholds. This loss of information is one disadvantage inherent to label based fusion.

The second assumption inherited from prior research is that the classification systems are combined using label fusion. In label fusion, the fusion of classification systems occurs after both systems have produced their own label sets. These label sets are then combined using some combination of logical Boolean rules or other rules to produce a combined label set. There are other methods of fusion. Another type of fusion is feature fusion, where a decision is made with respect to analysis of target features (i.e. Does the vehicle have wheels or tracks?) and a decision is made using data from both systems. With respect to Boolean rules, only the logical *AND/OR* rules will be investigated in this thesis. The derivation of ROC curves and therefore the derivation of bias formulas are done with respect to label-fusion under these two rules. This does limit the concept of bias currently to single applications of Boolean rules, but considering they are easy to implement and used heavily in research, this limitation is minimal.

A third assumption is the notion of conditional independence of classification systems with respect to the non-target partition of the event set. ROC curve formulas for the fused

classification system had been developed previously in a manner such that the formula could be written in terms of the ROC curves of the individual classification systems. In this manner, the performance of the combined system could be determined from the performances of the individual systems. Hence, the need for further testing is reduced and the use of previous testing results, as from legacy data sets, is leveraged.

### *1.5 Research Implications*

The results from this research may be used to determine the difference in classification system performance between competing label-based fusion techniques in application. Most importantly, it may draw attention to the importance of choosing the correct type of label fusion rule when the partitioning of events is finer than either target type or non-target type. This has not been scrutinized closely by many active researchers. Bias formulas may be used to adjust previously fused classification systems to a corrected performance by reweighting target prevalence, adjusting target environment, and adjusting the level of correlation between individual classification systems. This is accomplished through a simple and cost effective way of simulating the performance of classification systems a computer program. The groundwork may also be expanded upon by other researchers to develop techniques for comparing the performance of systems that are not based on Boolean rules as well as relaxing the notion of conditional independence of classification systems with respect to the non-target partition.

### *1.6 Outline of Sections*

In section II, a brief literature review outlines the research conducted into the area of label based fusion up to the current point in time. In section III, the underlying theory of classification system fusion and *across* and *within* label based fusion is derived. In

section IV, a brief outline of the computer simulation will be given. Finally, in section V the formulas for bias between competing *within* and *across* label-fused systems is derived and the simulation data is analyzed.

## *II Literature Review*

As budgets tighten and access to relatively inexpensive and efficient computing resources continues to rise, researchers are continuously searching for ways to improve the performance of autonomous classification systems. Generally the classification system being utilized is composed of an ensemble of classification systems that is then fused together using some sort of decision rule. As outlined in the introduction of (*Liggins et al: 1997)*, "fusion is necessary to integrate the data from different sensors and extract the relevant information on the targets." Traditionally, fusion occurred in a centralized architecture. That is, data from various sources were sent to a single location where the data was then fused in some way. Eventually this architecture evolved into a hierarchy of classification systems where low level systems process data and then send this information to more specialized classification systems to improve accuracy of decision. The means and manner in which information is being fused is rapidly changing to include a multitude of application specific methods.

There are many areas of active research where the fusion of information takes a lead role. For example, one of the more active areas of research involves the testing of hypotheses, and more specifically, either the identification or classification of an element of interest. In classification, information is fused to put an element of interest into one of a series of classes where common features are shared. Human face recognition and gender recognition are two areas where automated machine classification is of current interest (*Y. Pang et al: 2012)*. Identification is a more specific form of classification where the element can be physically labeled with regard to its true identity. Another example of active research with regard to information fusion is automatic target

recognition (ATR). In ATR research, the goal is to develop methods which permit the tracking and classification of objects through time and space. This has clear military and civilian applications, but is not the primary subject of this thesis. Further information on the subject can be gleaned from any number of sources such as (*Pilcher and Khotanzad: 2008),* (*Gross: 1999*), and (*Padgett and Woodward: 1997*).

*2.1 Fusion Methods*

There are a number of ways to fuse classification systems. Among the more prevalent methods in modern research are the use of neural networks, Boolean rules or voting rules, and statistical methods.

The fusion of classification systems or classification systems tends to occur most commonly on either the *features* of the data or on the *labels* produced by the system. In *feature fusion*, classification systems make decisions by analyzing the distinguishing attributes of those elements in question. In *label fusion,* the fusing of information occurs after the individual classification systems have already given a class label to the element in question. These two fusion approaches are arguably the most common in literature.

The use of neural networks is ubiquitous in literature (*Sinha*, *Gupta, Rao: 2001)*, (*Sani et al: 2009*), (*Werbos: 1991*), (*Won, Cho: 2003*),etc. Generally speaking, "a neural network conducts an analysis of the information and associates a probability estimate that the data matches the characteristics it has been trained to classify." (*Sani et al: 2009*). This training is done by modifying the thresholds of the system so that the neural network returns the classifications described by the end user. Neural networks are popular because they are relatively simple to design, make few assumptions regarding the underlying

distributions of elements it is tuned to classify, and is highly adaptable through the training process.

Boolean rules make use of Boolean algebra to classify elements of interest. Two of the simplest Boolean rules that are used heavily in research are the *AND* (conjunctive) rule and the *OR* (disjunctive) rule. These rules can be combined in various ways to create new decision or voting rules. Popular voting rules include the *majority vote* rule and the *sensor dominance* rule. The *majority vote* rule considers all permutations of the application of the *AND* rule to the classification systems and then applies the *OR* rule between each combination. The *sensor dominance* rule permits a single classification system to dominate the decision for the fused classification systems (*Schubert: 2005*). Some novel optimization approaches for receiver operating characteristic (ROC) curves have been proposed using a set of Boolean functions. The ROC curve is a graphical tool for depicting true positive rate along the vertical axis (hit rate) as compared to false positive rate along the horizontal axis (false alarm rate). This $BC_{ALL}$ approach developed by (*W. Khreich et al: 2012*). Applies a set of a 10 Boolean rules to two ROC curves and chooses the rule that optimizes performance at each point on the curve. Most Boolean rules that are used in the fusion of classification systems must have their performances determined after the actual fusion process. The work of (*Oxley M.E., Bauer, K.W.:2003*) and earlier papers began the notion of being able to describe the performance of fused classification systems prior to any sort of formal testing. This was done by deriving an expression for the performance of the combined system using properties of the performances of the individual classification systems. Oxley and Bauer originally developed an expression for the logical *OR* rule assuming independence between the

individual classification systems. The work of (*Schubert: 2005*) and (*Schubert et al: 2004*) has extended this work considerably, developing expression for the logical *AND* rule and developing the concepts of *within* and different types of *across* label fusion. This thesis relies on the theory developed in these fusion methods. In these works, the performance of the fused classification systems are expressed through the use of the ROC curve using the assumption of a two class label set.

Finally, there is the use of statistical modeling in fusion research. Generally this takes the form of some type of generalized linear model (GLM). A generalized linear model is a linear regression model that allows the response variables to be non-normally distributed. However, it is most common in literature to assume that the error or noise in statistical fusion is Gaussian or normally distributed. Though it is uncommon to work with non-Gaussian statistical models, industrial noise often shows non-Gaussian characteristics (*Niu, Zhu, Gu, Chu: 2009*). In statistical modeling, the goal is to approximate the distribution of those elements of interest generally through the use of least squares analysis or the method of maximum likelihood. Some advantages of statistical fusion are that modern software packages make this analysis very easy to conduct. A statistical model can be rigid in the fact that most linear regression techniques require that error terms be normal, independent, and identically distributed.

### *2.2 Independence and its Effects on Fusion*

Some methods described above make no assumptions regarding the level of dependence between classification systems. Most neural networks do not make assumptions regarding the level of dependence between elements being classified or between individual networks as any level of dependence present can be accounted for in

the training process. Dependence between classification systems has been one of the hurdles to the development of mathematical expressions for the performance of classification systems. In the works of Oxley and Bauer, it was assumed that the individual classification systems to be fused (and therefore their ROC curves) are statistically independent. This allows for the Boolean fusion rules to be defined. In probability theory, if two events are independent, then the probability of both events occurring simultaneously is equal to the product of the probabilities of the individual events occurring, that is, the occurrence of one or a set of events does not affect the outcome or occurrence of another event or set of events. If the two events are not independent, then calculating the probability of both events occurring simultaneously may either not be tractable or it may be very difficult. The Boolean *AND* rule for fusing individual classification systems takes the form of the independent *AND* rule in probability. Analogously, the Boolean *OR* rule takes the form of the independent *OR* rule in probability; that is, the sum of the probability of events A and B minus the product of the probability of events A and B.

When fusing ROC curves using label based fusion as presented in (*Schubert: 2005*), it is further required that the false positive values (the probability of false alarm rate) of the fused classification system be *conditionally independent* with respect to the non-target partition of the event set. Conditional independence of events is a stronger assumption than simply independence of events. As defined in (*Dawid: 1979)*, two random variables X and Y are conditionally independent with respect to a third random variable, Z, if the probability of X and Y given Z is the product of the probability of X given Z and the probability of Y given Z. More importantly, this should imply that the

probability of X given Y and Z is equivalent to the probability of X given Z. That is to say, any information about the random variable Y is superfluous and has no impact on the probability of X given Z. Dawid cautions that the use of *improper distributions/random variables* can lead to erroneous results. It may be possible in this instance to factor the conditional probability of X and Y given Z into the product of individual conditional probabilities, but the probability of X given Y and Z is no longer equal to the probability of X given Z. This is known as the *marginalization paradox*. This is important to discuss here as the research of this thesis along with the work of Schubert, Oxley, and Bauer requires the use of proper random variables. If this is not the case, this can lead to erroneous results.

The notions of independence between label-fused ROC curves were eased in the work of (*Schubert: 2005*) through the derivation of an expression for the correlation. This is a unique approach in that the level of dependence between classification systems can be calculated through the use of formulas involving only the ROC curves. Using this concept, functions were derived that accounted for a fixed level of correlation and hence independent label-fused ROC curves were those functions where the correlation coefficient, $\rho$, is zero.

Finally, there has been some research into the effects of correlation on the performance of classification systems. In (*Petrakos, Kaanelopoulous, Benediktsson, Pesaresi: 2000*) the researchers investigated the effects of correlation on fusion of classification systems using satellite imagery data. They did this by assessing the measure of agreement between different classification methods. In the work of (*Won, Cho: 2003*) the researchers selected ideal features from DNA microarray data that were negatively

correlated in an attempt to boost the performance of classification methods. This demonstrates that the presence of dependence can actually be used to enhance the performance of fused classification systems.

*2.3 Importance of Deriving a Method for Quantifying Bias in Label Fusion*

Classically speaking, the term *bias* in regards to classification theory generally implies that the data being classified is unequally weighted or that a subset of classes is more heavily favored. A good example of this comes from (*Abiantun and Savvides: 2009),* where the researchers refine the *Adaboost* algorithm to offset the bias inherently built into facial feature classification systems. Up to this point, most, if not all, research into the effects of imbalanced and biased classification systems has been done with respect to feature fusion. Given the derivation of a mathematical expression for quantifying both the performance and correlation of classification systems as provided by (*Schubert: 2005)* and (*Schubert, Oxley, and Bauer: 2005)*, it has become possible to characterize the inherent bias that exists in label fusion. The purpose of this thesis is to form a method and an algorithm to both quantify and adjust for the bias that exists between different types of label-fused classification systems with respect to event set partitions. This is important for the following reasons. To date, there has been almost no investigation into the effect of event set partitions on the performance of the fused system. Secondly, any bias that does exist between different label fusion methods could possibly be used as tools to tweak the performance of label-fused systems.

## *III Methods and Theoretical Development*

The mathematical theory of classification systems and Boolean fusion rules will be developed here. This will be achieved by developing the notion of conditional probability to describe the performance of a classification system, the receiver operating characteristic (ROC) curve to evaluate classification system performance, and defining the *across* and *within* label fusion rules. Further detail for much of these methods and theoretical development can be found in (*Schubert, et al: 2005)* and *(Schubert: 2005)*.

### *3.1 Classification Systems*

#### *3.1.1 Single Classification System*

Define $\Gamma$ to be a population set of outcomes, i.e., the overarching or underlying event set. Let $G$ be a σ-algebra of subsets of $\Gamma$. Then, $(\Gamma, G)$ defines a measurable space (*Schubert: 2005*). Define $P_\Gamma$ to be a probability measure defined on $G$. This implies that $(\Gamma, G, P_\Gamma)$ defines a probability measure space. Let $s$ be a sensor that maps outcomes from $\Gamma$ to a new data set, $\Delta$. Let $D$ be a σ-algebra on $\Delta$, implying that $(\Delta, D)$ defines a measurable space. Furthermore, define $P_\Delta$ to be a probability measure defined on $D$; therefore $(\Delta, D, P_\Delta)$ defines a probability measure space. Some examples of data sets may include different segments of the electromagnetic spectrum, search engine keywords, or images. Sometimes this data is too nebulous to make an accurate decision, so another mapping $p$ (a processor) is defined on $\Delta$ that can be used to produce an object $f$ which is called a feature. Let $\Phi$ be a feature set and define $F$ to be a σ-algebra of subsets of $\Phi$. This makes $(\Phi, F)$ a measureable space. Letting $P_\Phi$ be a probability measure defined on $F$ defines the probability measure space $(\Phi, F, P_\Phi)$. In most circumstances, including this thesis, the feature $f$ is a vector of real numbers, though it need not be. Let $\Theta$ be a set of

parameters. For each $\theta \in \Theta$, let $a_\theta$ be a classification system mapping that takes elements of $\Phi$ into $\Lambda$, the label set. Defining $L$ to be a σ-algebra on $\Lambda$ makes $(\Lambda, L)$ a measureable space. Letting $P_\Lambda$ be a probability measure defines a probability measure space $(\Lambda, L, P_\Lambda)$. The label may take any number of forms, but in this thesis, elements from $\Lambda$ will take the form {non-target, target} ≅ {n, t}, depending on context. The composition of these mappings creates the single classification system, defined in this context to be $A_\theta$. That is, for every $\theta \in \Theta$:

$$A_\theta = a_\theta \circ p \circ s \tag{3.1}$$

$$\Gamma \xrightarrow{A_\theta} \Lambda$$

### *3.1.2 Multiple Classification Systems*

It is possible to combine two or more classification systems together. In this thesis, only the fusion of two classification systems is considered. Further, the systems to be combined are fused together using label fusion. Label fusion may be loosely defined as the joint system decision based on processing the labels (the decisions) given to elements in the event set by individual classification systems. Consider two classification systems: $A_\theta$ and $B_\pi$. Let $A_\theta$ be the system defined previously. Let system $B_\pi$ be defined as the classification system defined by the composition mapping:

$$B_\pi = b_\pi \circ p_2 \circ s_2 \tag{3.2}$$

Classifier $b_\pi$ maps elements from the feature set associated with system $B_\pi$ into its label set. Note that the sensor, $s_2$, and processor, $p_2$, are different from those processors and

sensors defined for classification system $A_\theta$. Now, consider that these two classifications systems observe the same element, *x*, in the event set, Γ. That is, for $x \in \Gamma$:

$$A_\theta = a_\theta \circ p_1 \circ s_1(x) \qquad (3.3)$$

$$B_\pi = b_\pi \circ p_2 \circ s_2(x)$$

The labels produced by these two compositions are then fused to create a system label as generated by both classification systems, hence label fusion. The following schematic outlines this process of label based fusion.

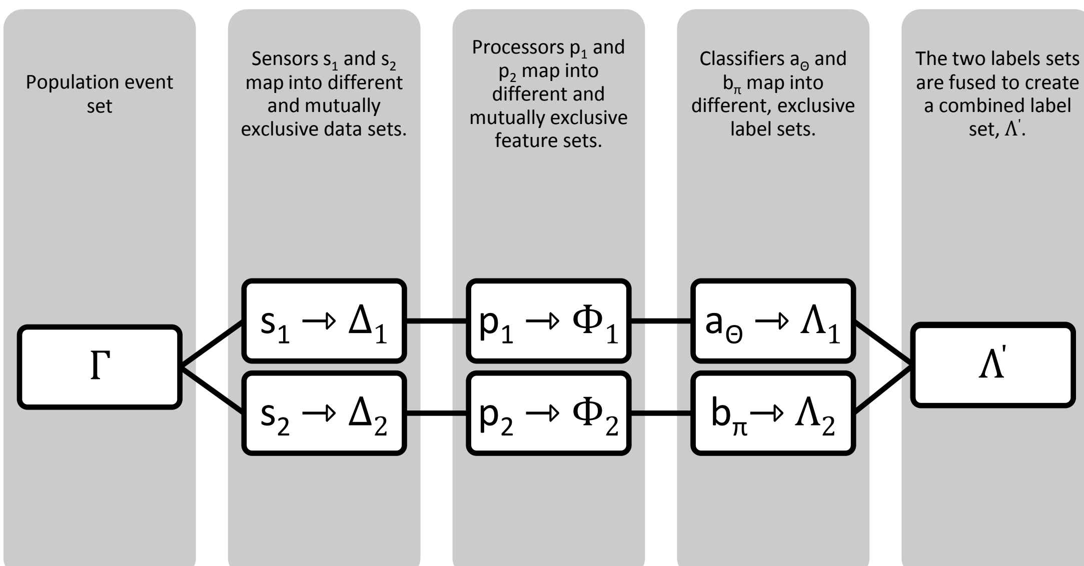



**Figure 3-1 - Mapping of element from the population of event sets, Γ, into the fused label set, Λ'.**

It is possible to develop other system mappings. Say for example that the two data sets from above are simultaneously mapped to the same feature set.

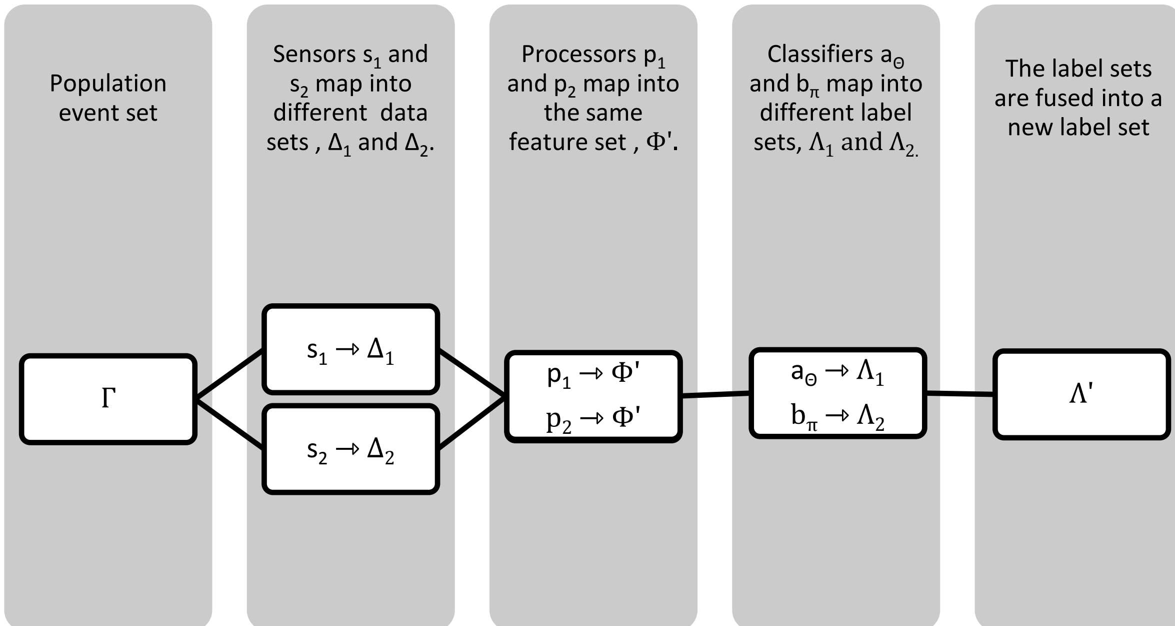



**Figure 3-2 - Alternate mapping of elements from Γ into Λ'. Note here that $p_1$ and $p_2$ map into the same feature set, Φ'.**

The act of fusing information at an earlier stage (fusing into the same feature set) may change the way that the classification systems label events. This is largely a question of design on the part of the experimenter and the environment in which the classification system is to be applied. Other types of mapping exist, but this thesis is concerned with those systems $A_\theta$ and $B_\pi$ that map into fused label sets.

### *3.1.3 Properties of Classification System Mappings*

Before developing a formulaic approach to analyze the data returned by the classification system, a few properties must be outlined here.

**Definition 3.1** *(Pre-image) Let X and Y be nonempty sets. Let* f *be a mapping that takes an element $x \in X$ into Y. Given some subset $Y \subset Y$, the pre-image of* f *is defined to be the subset in X such that*

$$f^{\natural}[Y] = \{x \in X : f(x) \in Y\}$$

*Hence, the pre-image of a subset* $\mathbb{Y} \subset Y$ *are those elements in X that are mapped by* f *into* $\mathbb{Y}$.

In most mathematical texts, the pre-image is generally considered synonymous with the *inverse image*. Considering that the mapping *f* described above need not be invertible, the natural symbol (♮) as used by (*Schubert, C., Oxley, M. E., Bauer, K: 2005)* will be used here. The pre-image for a classification system allows a way to map backwards from the label set into some subset or element from the population event set, Γ. That is, the image and pre-image for $A_\theta$ can be written respectively as follows.

$$\begin{aligned} A_\theta &= a_\theta \circ p \circ s \\ A_\theta^\natural &= a_\theta^\natural \circ p^\natural \circ s^\natural \end{aligned} \tag{3.4}$$

The performance associated with a certain classification system is assessed using probability theory. Therefore, it will be necessary to develop the notion of a measurable space and probability measure space so that this may be done.

**Definition 3.2** *(Measurable Mapping) Let Ξ be a σ-algebra of subsets of set X. Let Ψ be a σ-algebra of subsets of set Y. This implies that (Ξ, X) and (Ψ, Y) are measurable spaces. A mapping* f *is measurable if for every subset* $\psi \in \Psi$

$$f^\natural[\Psi] = \{\psi \in \Psi : f^\natural(\psi) \in \Xi\}$$

*That is, the pre-image* $f^\natural(\psi) \in \Xi$.

Using this definition, the composition mapping described by $A_\theta^\natural$ is measurable. Thus, $A_\theta^\natural: L \rightarrow D$. That is, the mapping takes the subsets from the σ-algebra associated with the label set, Λ, into the σ-algebra associated with the data set, Δ. It is implied that the composition of measurable mappings must take the subsets from the σ-algebra associated with Λ into the σ-algebra associated with the feature set, Φ, before mapping back into the

σ-algebra of the data set. Furthermore, if it is assumed that the composition is measurable, a *random variable* may be defined for the system in question. Consider the measurable mapping defined by **G** = p(s). This mapping is a called a random element of the feature set, Φ, and is called a *random variable* when (Φ, *F*) = (ℝ, *F*(ℝ)) (*Schubert: 2005)*.

**Definition 3.3 –** *(Measurable Mapping) Let (Γ, G) and (Φ, F) be measurable spaces.* **M** *is called a random mapping if* **M**: *Γ → Φ is a Γ-Φ measurable mapping.*

**Definition 3.4 –** *(Induced Probability Distribution) Let $P_\Gamma$ be a probability measure defined on G such that (Γ, G, $P_\Gamma$) defines a probability measure space. Further, suppose that (Φ, F) defines a measurable space. Let* **M**: *Γ → Φ be a random mapping. Define the set function* $[P_\Gamma \circ M]^\natural$ *on F*

$$[P_\Gamma \circ \boldsymbol{M})]^\natural(f) = P_\Gamma(\boldsymbol{M}^\natural(f))$$

*for every f ∈ F. Thus,* $P_\Phi = P_\Gamma \circ \mathbf{M}^\natural$ *defines a probability measure on (Φ, F) known as the induced probability/distribution measure of* **M** *and (Φ, F, $P_\Phi$) is a probability measure space.*

The concept and formulation of measurable mappings is necessary in order to analyze classification systems. Further information and technical development may be found in (*Schubert: 2005)*. Measurable mappings and more importantly, probability distributions and measure spaces, allows a way to assign probability outcomes that can be mapped back to the original event sets. This is clearly important given that under most circumstances, the original event is unknown. In this thesis, these concepts will be used

to develop a way to quantify the difference in performance between *within* and *across* label fusion.

*3.2 Performance of Classification Systems*

After the classification system of interest has assigned labels to the elements in the population event set, it is still unknown how well the system has performed. It is highly unlikely that the classification system has worked exactly as intended, i.e., made perfect classifications. Therefore, there must be a way to quantify the errors and successes.

The receiver operator characteristic curve allows a way to qualitatively and quantitatively assess classification system performance. The ROC curve is useful as it graphically depicts the difference between false alarm rate and true hit rate for each parameter threshold of the classification system. The classification system produces a true positive (TP) when it labels an element from the target population event set to which it is tuned as a "target." The classification system produces a false positive (FP) when it labels an element from the non-target population event set as "target." A ROC curve is created by graphing FP rates along the horizontal (X) axis and TP rates along the vertical (Y) axis. The classification system approximates the true ROC curve using empirical data produced during the composition of mappings.

There are four possible probability outcomes given that an event can receive a target or non-target label and the event set is partitioned into target and non-target subsets. This is valid given that a probability measure space has been defined. Consider system $B_\pi$. Note that $\Gamma$ is composed of all subsets of targets and non-targets. For the sake of simplicity, assume that $\Gamma$ is divided into two populations. Denote the event set

composed of targets system $B_\pi$ is tuned to classify as $\Gamma_t$. Denote the event set composed of non-targets system $B_\pi$ is tuned to identify as $\Gamma_n$. Let $P_{TP}(B_\pi)$ denote the probability that $B_\pi$ correctly labels an element $t \in \Gamma_t$ as a "target." This is the definition of a true positive classification by system $B_\pi$ as the system correctly labeled a target with the target label. Mathematically, this can be modeled using conditional probability:

$$P_{TP}(\mathrm{B}_\pi) = P\{B_\pi(t_B) = t : t_B \in \Gamma_t\} = \frac{P(B_\pi^\natural(L_t) \cap \Gamma_t)}{P(\Gamma_t)} \qquad (3.5)$$

Let $P_{FP}(B_\pi)$ denote the probability that $B_\pi$ incorrectly labels an element $n \in \Gamma_n$ as a "target." This is a measure of a false positive classification by system $B_\pi$ as the system falsely labeled the non-target element with a target label. This too may be modeled using conditional probability

$$P_{FP}(\mathrm{B}_\pi) = P\{B_\pi(n_B) = t : n_B \in \Gamma_t\} = \frac{P(B_\pi^\natural(L_t) \cap \Gamma_n)}{P(\Gamma_n)} \qquad (3.6)$$

Next, let $P_{TN}(B_\pi)$ denote the probability that $B_\pi$ correctly labels an element $n \in \Gamma_n$ as a "non-target." This is the definition of a true negative classification by system $B_\pi$ as the system correctly labeled a non-target with the non-target label. Mathematically speaking:

$$P_{TN}(B_\pi) = P\{B_\pi(n_B) = n : \ n_B \ \in \ \Gamma_n\} = \frac{P(B_\pi^\natural(L_n) \cap \Gamma_n)}{P(\Gamma_n)} \qquad (3.7)$$

Lastly, let $P_{FN}(B_\pi)$ denote the probability that $B_\pi$ incorrectly labels an element $t \in \Gamma_t$ as a "non-target." Then $P_{FN}(B_\pi)$ is the probability that $B_\pi$ produced a non-target label

for an element that was, in truth, a target. This is more commonly known as a false negative designation and is described mathematically as follows:

$$P_{FN}(B_\pi) = P\{B_\pi(t_B) = n :\ t_B \in \Gamma_n\} = \frac{P(B_\pi^\natural(L_n) \cap \Gamma_t)}{P(\Gamma_t)} \tag{3.7}$$

These definitions lead to two important properties involved with these probability statements.

$$\begin{aligned} P_{TP}(\mathrm{B}_\pi) + P_{FN}(\mathrm{B}_\pi) &= \frac{P\left(B_\pi^\natural(L_t) \cap \Gamma_t\right)}{P(\Gamma_t)} + \frac{P\left(B_\pi^\natural(L_n) \cap \Gamma_t\right)}{P(\Gamma_t)} \\ &= \frac{P\left(B_\pi^\natural(L_t) \cap \Gamma_t\right) + P\left(B_\pi^\natural(L_n) \cap \Gamma_t\right)}{P(\Gamma_t)} \\ &= \frac{P\left(\left(B_\pi^\natural(L_t) \cup B_\pi^\natural(L_n)\right) \cap \Gamma_t\right)}{P(\Gamma_t)} \\ &= \frac{P(\Gamma_t)}{P(\Gamma_t)} = 1 \end{aligned} \tag{3.8}$$

$$\begin{aligned} P_{FP}(\mathrm{B}_\pi) + P_{TN}(\mathrm{B}_\pi) &= \frac{P\left(B_\pi^\natural(L_t) \cap \Gamma_n\right)}{P(\Gamma_n)} + \frac{P\left(B_\pi^\natural(L_n) \cap \Gamma_n\right)}{P(\Gamma_n)} \\ &= \frac{P\left(B_\pi^\natural(L_t) \cap \Gamma_n\right) + P\left(B_\pi^\natural(L_n) \cap \Gamma_n\right)}{P(\Gamma_n)} \\ &= \frac{P\left(\left(B_\pi^\natural(L_t) \cup B_\pi^\natural(L_n)\right) \cap \Gamma_n\right)}{P(\Gamma_n)} \\ &= \frac{P(\Gamma_n)}{P(\Gamma_n)} = 1 \end{aligned} \tag{3.9}$$

Note that for any of the above probabilities, all are dependent upon the parameters of the classification system. A single probability value is associated with a specific parameter. That is to say, for each parameter combination, a TP, FP, TN, and FN is associated with the given combination. These probabilities change as the parameter values change. Given that the goal is to produce a ROC curve for the classification

system $B_\pi$, the focus will be on the TP and FP values. Define π to be the parameter set for system $B_\pi$, then the following set of ordered triples defines the *trajectory* of $\mathcal{B}$.

$$T_{\mathcal{B}} = \left\{\left(\pi, P_{FP}(B_\pi), P_{TP}(B_\pi)\right) : \pi \in \Pi \right\}$$

Projecting the trajectory onto the FP and TP components yields those points which define the frontier (denoted by $\mathcal{F}$) of the ROC curve.

$$\mathcal{F}_{\mathcal{B}} = \left\{\left(P_{FP}(B_\pi), P_{TP}(B_\pi)\right) : \ \pi \in \Pi \right\}$$

Assuming that the set π is homeomorphic to $\mathbb{R}$, then the trajectory corresponds to a curve in $\mathbb{R}^3$ and the frontier corresponds to a projection onto $\mathbb{R}^2$ (*Schubert: 2005)*. This frontier is the ROC curve for system $B_\pi$. There may be other points that empirically exist on the curve, but the definition of a ROC curve requires that the ROC function be non-decreasing. Considering that both $P_{FP}(B_\pi)$ and $P_{TP}(B_\pi) \in [0,1]$, $\mathcal{F}_{\mathcal{B}}$ is a projection onto [0,1] x [0,1], the unit square. In this instance the parameter set is one-dimensional; therefore $\mathcal{F}_{\mathcal{B}}$ is composed of a single curve that is projected onto $\mathbb{R}^2$. The same is true for classification system $A_\theta$. Next, the ROC curve and ROC function is formally defined.

**Definition 3.5** *(ROC curve) Assume that* $\mathcal{B} = \{B_\pi: \pi \in \Pi\}$ *defines the classification family of interest. Let* $p \in [0, 1]$ *correspond to the value of the false positive. Similarly, let* $q \in [0, 1]$ *correspond to the value of the true positive. The ROC curve for* $\mathcal{B}$ *is defined as*

$$\mathcal{F}_{\mathcal{B}} = \left\{(p, q) : p \ \in \ [0, 1] \ and \ q = max\{P_{TP}(B_\pi) : \pi \in \Pi \ \ and \ P_{FP}(B_\pi) \le p \}\right\}$$

*and the corresponding ROC function may be defined as*

$$\mathcal{F}_{\mathcal{B}}(p) = \ max\{P_{TP}(B_\pi) : \ \pi \in \Pi \ and \ P_{FP}(B_\pi) \le p \}$$

*3.3 Across Label Fusion and Within Label Fusion*

Now that a method is in place to create the ROC curve for individual classification systems, it is important to consider how to fuse these systems in an attempt to increase predictive power. In order to use the performance of the individual classification systems to quantify the performance of the fused system, there must be some accounting for the different target and non-target sets that the classification systems are tuned to label. Some formal definitions regarding how the classification systems partition the event set has been developed and will be briefly touched upon here.

**Definition 3.6** *(Finite Partition) Let S be a non-empty set.*

1) *If* $A_m \cap A_n = \emptyset \ \forall m = 1, \ldots, M$ *and* $n = 1, \ldots, N; m \neq n; M, N < \infty$ *(Pairwise disjoint)*
2) $\cup_{i=1}^{\infty} A_i = S$

*Let* $\{A_1, \ldots, A_k, \ldots\}$ *be a countable collection of subsets of S. Then* $\{A_1, \ldots, A_k, \ldots\}$ *forms a finite partition of S.*

**Definition 3.7** *(True Partition) Assume the following*

1) $(\Gamma, G)$ *is a measurable space*
2) $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_M\}$ *is a finite label set.* $\boldsymbol{\Lambda}$ *is the power set of* $\Lambda$ *such that* $(\Lambda, \boldsymbol{\Lambda})$ *is a measurable space.*
3) $\boldsymbol{t} : \Gamma \rightarrow \Lambda$ *is a measurable mapping . The domain of* $\boldsymbol{t}$ *is* $\Gamma$ *and the range of* $\boldsymbol{t}$ *is* $\Lambda$.

*Then* $\boldsymbol{t}$ *defines a truth mapping. The collection of pre-images* $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_N\}$ *defined by* $\Gamma_n = \boldsymbol{t}^{\natural}(\{\lambda_n\}) \in G \ \forall n = 1, \ldots, N$ *forms a partition of* $\Gamma$ *that is the true partition with respect to the truth mapping,* $\boldsymbol{t}$. *(Schubert: 2005)*

**Definition 3.8** *(Within-Fusion Rule) Assume the following*

1) *$(\Gamma, G)$ is a measurable space.*
2) *$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_M\}$ is a finite label set. $\boldsymbol{\Lambda}$ is the power set of $\Lambda$ such that $(\Lambda, \boldsymbol{\Lambda})$ is a measurable space.*
3) *$G_\Lambda = \left\{\Gamma_{\lambda_1}, \Gamma_{\lambda_2}, \ldots, \Gamma_{\lambda_M}\right\} \subset G$ is the truth partition of $\Gamma$ with respect to $\Lambda$.*

   *If the classification systems $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_N : \Gamma \rightarrow \Lambda$ are measurable mappings designed to map $\Gamma_{\lambda_M} \rightarrow \lambda_M$ for each $m = 1, \ldots, M$, then the fusion rule $r$ that combines the collection of classification system systems yielding the new classification system*

$$\mathcal{B}_0 = r(\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_N)$$

   *is said to be a within-fusion rule.* (*Schubert: 2005)*

**Definition 3.9** *(Across-Fusion Rule) Assume the following*

1) *$(\Gamma, G)$ is a measurable space.*
2) *$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_M\}$ is a finite label set. $\boldsymbol{\Lambda}$ is the power set of $\Lambda$ such that $(\Lambda, \boldsymbol{\Lambda})$ is a measurable space.*
3) *$G_\Lambda = \left\{\Gamma_{\lambda_1}, \Gamma_{\lambda_2}, \ldots, \Gamma_{\lambda_M}\right\} \subset G$ is the truth partition of $\Gamma$ with respect to $\Lambda$.*
4) *$\boldsymbol{\Lambda}^{(0)}, \boldsymbol{\Lambda}^{(1)}, \ldots, \boldsymbol{\Lambda}^{(N)} \subset \boldsymbol{\Lambda}$ are partitions of $\Lambda$.*
5) *For each $n = 0, \ldots, N$, the integer $M^{(n)} = card(\boldsymbol{\Lambda}^{(n)}) \leq M$, and the partition $\boldsymbol{\Lambda}^{(n)}$ is congruent to the label set $\Lambda^{(n)} = \left\{w_1^{(n)}, \ldots, w_{M^{(n)}}^{(n)}\right\}$*
6) *For each $n = 0, \ldots, N$, the partition $G^{(n)} \subset G$ is the true partition of $\Gamma$ with respect to $\boldsymbol{\Lambda}^{(n)}$.*

*If the classification systems $\mathcal{B}_1 : \Gamma \nrightarrow \Lambda^{(1)}$, $\mathcal{B}_2 : \Gamma \nrightarrow \Lambda^{(2)}$, …, $\mathcal{B}_N : \Gamma \nrightarrow \Lambda^{(N)}$ are designed to map each event set $\Gamma_w \in G^{(n)}$ to the corresponding $w \in \Lambda^{(n)}$, then for every n = 1, 2, …, N, the fusion rule r that combines the collection of classification systems yielding the new classification system system*

$$\mathcal{B}_0 = r(\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_N)$$

*is said to be an across-fusion rule.* (*Schubert: 2005)*

For the sake of clarification***, a within label-fused system is composed of individual classification systems which are both tuned to the same target and non-target partitions of the event set. An across label-fused system is composed of individual classification systems that are tuned to different partitions of the target and non-target event set.***

*3.3.1 Within Label Fusion*

Assume there exists two classification systems, $A_\theta$ and $B_\pi$, which are to be fused under *within* label fusion. Further, assume that classification systems $A_\theta$ and $B_\pi$ are both designed to classify a single target type and a single non-target type. Thus, the label set in question is $\Lambda = \{t, n\}$ where t denotes "target" and n denotes "non-target." Furthermore, $G_\Lambda = \{\Gamma_t, \Gamma_n\} \subset G$ is the true partition of $\Gamma$ with respect to $\Lambda$. Hence, system $A_\theta : \Gamma \nrightarrow \Lambda$ where $A_\theta$ is designed to map $\Gamma_t \in G_\Lambda$ to the corresponding $t \in \Lambda$ and $\Gamma_n \in G_\Lambda$ to the corresponding $n \in \Lambda$. The partitions of the event set and label sets of systems $A_\theta$ and $B_\pi$ are the same. That is, for any target element, systems $A_\theta$ and $B_\pi$ will label the element using the same "t" label.

*3.3.2 Across Label Fusion*

There are three ways that classification systems may be categorized as *across* label fusion in this thesis as previously outlined by (*Schubert: 2005).*

1) Case I – Each classification system labels mutually disjoint targets.

2) Case II – One classification system labels a subset of targets of the other classification system.

3) Case III – The targets of the two classification systems overlap, creating a subset of targets labeled by both systems.

*Case I – Each classification system labels mutually disjoint targets.*

Assume classification system $A_\theta$ is designed to classify target type 1 ($t_1$) and classification system system $B_\pi$ is designed to classify target type 2 ($t_2$). Thus, the label set of interest, $\Lambda = \{t, n\}$ where $t = t_1 \cup t_2$ and $t_1 \cap t_2 = \emptyset$. $G_\Lambda = \{\Gamma_{t_1}, \Gamma_{t_2}, \Gamma_n\} \subset G$ is the true partition of $\Gamma$ with respect to $\Lambda$. The label set for classification system $A_\theta$ is $\Lambda^{(A)} = \{t_1, n_1\}$ and the label set for classification system $B_\pi$ is $\Lambda^{(\mathcal{B})} = \{t_2, n_2\}$. Note that $n_1$ denotes the complementary non-target type outcome that is composed of both "n" and "$t_2$" elements. Similarly, $n_2$ denotes the complementary non-target outcome that is composed of both "n" and "$t_1$" elements. Hence, $A_\theta : \Gamma \rightarrow \Lambda^{(A)}$ where $A_\theta$ is designed to map $\Gamma_{t_1} \in G^{(A)}$ to the corresponding $t_1 \in \Lambda^{(A)}$ and $\Gamma_{n_1} \in G^{(A)}$ to the corresponding $n_1 \in \Lambda^{(A)}$. Analogously, $B_\pi : \Gamma \rightarrow \Lambda^{(\mathcal{B})}$ where $B_\pi$ is designed to map $\Gamma_{t_2} \in G^{(\mathcal{B})}$ to the corresponding $t_2 \in \Lambda^{(\mathcal{B})}$ and $\Gamma_{n_2} \in G^{(\mathcal{B})}$ to the corresponding $n_2 \in \Lambda^{(\mathcal{B})}$.

*Case II – One classification system labels a subset of targets of the other classification system.*

Assume classification system $A_\theta$ is designed to classify $t_1$ events and assume classification system $B_\pi$ is designed to classify any target type (t). Thus, the label set of interest, $\Lambda = \{t, n\}$. $G_\Lambda = \{\Gamma_{t_1}, \Gamma_{t_2}, \Gamma_n\} \subset G$ is the true partition of $\Gamma$ with respect to $\Lambda$. The label set for classification system $A_\theta$ is $\Lambda^{(A)} = \{t_1, n_1\}$ and the label set for classification system $B_\pi$ is $\Lambda^{(\mathcal{B})} = \{t, n\}$. Note that $n_1$ denotes the complementary non-target type outcome that is composed of both "n" and "$t_2$" elements. Hence, $A_\theta : \Gamma \rightarrow \Lambda^{(A)}$ where $A_\theta$ is designed to map $\Gamma_{t_1} \in G^{(A)}$ to the corresponding $t_1 \in \Lambda^{(A)}$ and $\Gamma_{n_1} \in G^{(A)}$ to the corresponding $n_1 \in \Lambda^{(A)}$. Analogously, $B_\pi: \Gamma \rightarrow \Lambda^{(\mathcal{B})}$ where $B_\pi$ is designed to map $\Gamma_t \in G^{(\mathcal{B})}$ to the corresponding $t \in \Lambda^{(\mathcal{B})}$ and $\Gamma_n \in G^{(\mathcal{B})}$ to the corresponding $n \in \Lambda^{(\mathcal{B})}$.

*Case III – The targets of the two classification systems overlap, creating a subset of targets labeled by both systems.*

Assume classification system $A_\theta$ is designed to classify target types 1 and 2 ($t_1$ and $t_2$). Assume classification system $B_\pi$ is designed to classify target types 2 and 3 ($t_2$ and $t_3$). The label set is $\Lambda = \{t, n\}$ where $t = t_1 \cup t_2 \cup t_3$ and $t_1 \cap t_3 = \emptyset$. $G_\Lambda = \{\Gamma_{t_1}, \Gamma_{t_2}, \Gamma_{t_3} \Gamma_n\} \subset G$ is the true partition of $\Gamma$ with respect to $\Lambda$. The label set for classification system $A_\theta$ is $\Lambda^{(A)} = \{t_1, t_2, n_{12}\}$ and the label set for classification system $B_\pi$ is $\Lambda^{(\mathcal{B})} = \{t_2, t_3, n_{23}\}$. Note that $n_{12}$ denotes the complementary non-target composed of both "$t_3$" and "n" labels for system $A_\theta$. Similarly, $n_{23}$ denotes the complementary non-target type for system $B_\pi$ composed of both "$t_1$" and "n" elements. Hence, $A_\theta : \Gamma \rightarrow \Lambda^{(A)}$ where $A_\theta$ is designed to map $\{\Gamma_{t_1}, \Gamma_{t_2}\} \in G^{(A)}$ to the corresponding $t_{12} \in \Lambda^{(A)}$ and $\{\Gamma_n, \Gamma_{t_3}\} \in G^{(A)}$ to the corresponding $n_{12} \in \Lambda^{(A)}$. Analogously, $B_\pi: \Gamma \rightarrow \Lambda^{(\mathcal{B})}$ where $B_\pi$ is designed

to map $\{\Gamma_{t_2}, \Gamma_{t_3}\} \in G^{(\mathcal{B})}$ to the corresponding $t_{23} \in \Lambda^{(\mathcal{B})}$ and $\{\Gamma_n, \Gamma_{t_1}\} \in G^{(\mathcal{B})}$ to the corresponding $n_{23} \in \Lambda^{(\mathcal{B})}$.

### *3.4 Label Fusion Rules*

In this thesis, the focus is *within* and *across* label-fusion under the Boolean *AND* and Boolean *OR* rules.

#### *3.4.1 Boolean AND Label Fusion Rule*

The *AND* rule is a binary operation that is defined on the label set, Λ. This operator will be defined using the logical *AND* symbol, $\wedge$. It is defined in the following table.

| $\wedge$ | t | n |
|---|---|---|
| t | t | n |
| n | n | n |

**Table 3-1 - Truth table of label outcomes for Boolean *AND* rule.**

Now, consider classification systems $A_\theta$ and $B_\pi$. The Boolean *AND* label-fused classification system $C^{and}_{(\theta,\pi)}$ may be defined as the following.

$$C^{\text{AND}}_{(\theta,\pi)}(x) = A_\theta(x) \wedge B_\pi(x) \;\forall\; x \in \Gamma$$

That is to say, the fused classification system returns a target label (t) only when both classification systems $A_\theta$ and $B_\pi$ label the same element in question as being a target from the target population event set. The fused classification system returns a non-target label (n) when either $A_\theta$ or $B_\pi$ returns a non-target label or $A_\theta$ and $B_\pi$ both return a non-target label. The *AND* fusion rule is sometimes called a conservative label/decision rule because it requires that both individual classification systems label the element as a target in order to receive a combined target classification. Note that under the fusion rule it is not known which system returned the non-target label or whether both systems returned

the non-target label. The only information available is the combined decision made by the fused system. If knowledge of the feature set is available, implying that the true partition of the event sets is actually known, then it can be determined which system or systems returned the non-target label.

### *3.4.2 Boolean OR Label Fusion Rule*

The Boolean *OR* rule is also a binary operation defined on $\Lambda$. The *OR* operator will be defined using the logical *OR* symbol, $\vee$. It is defined in the following truth table:

| $\vee$ | t | n |
|---|---|---|
| t | t | t |
| n | t | n |

**Table 3-2 - Truth table of label outcomes for Boolean *OR* rule.**

Consider systems $A_\theta$ and $B_\pi$. The Boolean *OR* label-fused classification system is defined as the following

$$C^{OR}_{(\theta,\pi)}(x) = A_\theta(x) \vee B_\pi(x)\ \forall\ x \in \Gamma$$

Using the truth table as a reference, notice that the combined *OR* classification system labels an element $x \in \Gamma$ as a target if either one or both classification systems labels the element as a target. A non-target label can only occur when both individual classification systems label the element in question as a non-target. Similar to the *AND* fusion rule, without knowledge of the true partition of the event set, it is not known which classification system(s) labeled the element as a target.

### *3.4.3 Within AND/OR Label Fusion*

Now that the Boolean rules are in place, they can be applied to *within* label fusion. Consider first the *within AND* label fusion rule, which will henceforth be referred to using the shorthand $\wedge^w$. Recall that in *within* label fusion, systems $A_\theta$ and $B_\pi$ classify

the same target and non-target types. Let *t* denote the targets and *n* denote the non-targets that $A_{\theta}$ and $B_{\pi}$ are tuned to classify. Consider the following table that outlines the four possible outcomes for systems $A_{\theta}$ and $B_{\pi}$.

| Truth | $A_{\theta}$ | | $B_{\pi}$ | |
|---|---|---|---|---|
| | t | n | t | n |
| t | TP | FP | TP | FP |
| n | FN | TN | FN | TN |

Table 3-3 - Within ***AND*** label fusion for individual classification systems $A_{\theta}$ and $B_{\pi}$.

The corresponding probabilities for these outcomes as determined by label fusion are outlined below.

| $\wedge^{w}$ | $\Gamma_{t}$ (True target partition) | $\Gamma_{n}$ (True non-target partition) |
|---|---|---|
| $A^{\natural}_{\theta}(\{t\}) \cap B^{\natural}_{\pi}(\{t\})$ | TP | FP |
| $A^{\natural}_{\theta}(\{t\}) \cap B^{\natural}_{\pi}(\{n\})$ | FN | TN |
| $A^{\natural}_{\theta}(\{n\}) \cap B^{\natural}_{\pi}(\{t\})$ | FN | TN |
| $A^{\natural}_{\theta}(\{n\}) \cap B^{\natural}_{\pi}(\{n\})$ | FN | TN |

Table 3-4 - Within ***AND*** label fusion for fused classification system with respect to partitions of the event set.

As can be seen, the errors made by the individual classification systems add to the probability of a FN outcome for the fused classification system. However, the fused system determines true non-targets under the combined $\wedge^{w}$ label fusion rule well due to the large number of partitions for a correct TN outcome.

Next, consider the *within OR* label fusion rule which will be referred to using the shorthand notation $\vee^{w}$. Table 3.3 also applies to the *within OR* label fusion rule, but the outcomes with respect to the different target partitions is different under the *OR* rule.

| $\vee^{w}$ | $\Gamma_{t}$ (True target partition) | $\Gamma_{n}$ (True non-target partition) |
|---|---|---|
| $A^{\natural}_{\theta}(\{t\}) \cup B^{\natural}_{\pi}(\{t\})$ | TP | FP |
| $A^{\natural}_{\theta}(\{t\}) \cup B^{\natural}_{\pi}(\{n\})$ | TP | FP |
| $A^{\natural}_{\theta}(\{n\}) \cup B^{\natural}_{\pi}(\{t\})$ | TP | FP |
| $A^{\natural}_{\theta}(\{n\}) \cup B^{\natural}_{\pi}(\{n\})$ | FN | TN |

Table 3-5 - Within *OR* label fusion for combined classification system.

Notice that under the *within OR* rule that the fused system excels at correctly classifying targets in the environment. However, if either one or both classification systems incorrectly labels a non-target with a target label, this significantly adds to the FP probability.

### *3.4.4 Across AND/OR Label Fusion*

Recall that under *across* label fusion, the target and non-target event sets are partitioned in three different ways. For case I, classification system $A_\theta$ is tuned to $t_1$ and the complementary non-target $n_1$. Classification system $B_\pi$ is tuned to $t_2$ and the complementary non-target set composed of $t_1$ and n, that is, $n_2$. Symbolically, the *across I AND rule* will be defined as $\wedge^I$. The following truth table outlines the target and non-target designation of the $\wedge^I$ combined classification system.

| $\wedge^I$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_1$ | $n_1 = \{n,t_2\}$ |
| $t_2$ | t | n |
| $n_2 = \{n,t_1\}$ | n | n |

Table 3-6 – *Across I AND* truth table for fused system.

In terms of the four probability outcomes and their associated probability statements

| Truth under $\wedge^I$ rule | $A_\theta$ | | $B_\pi$ | |
|---|---|---|---|---|
| | $t_1$ | $n_1 = \{n, t_2\}$ | $t_2$ | $n_2 = \{n,t_1\}$ |
| $t_1$ | TP | FN | FP | TN |
| $t_2$ | FP | TN | TP | FN |
| n | FP | TN | FP | TN |

Table 3-7 - Truth table for individual classification systems under *across I AND* label fusion.

| $\wedge^I$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|
| $A^{\natural}_{\theta}(\{t_1\}) \cap B^{\natural}_{\pi}(\{t_2\})$ | TP | TP | FP |
| $A^{\natural}_{\theta}(\{t_1\}) \cap B^{\natural}_{\pi}(\{n_2\})$ | FN | FN | TN |
| $A^{\natural}_{\theta}(\{n_1\}) \cap B^{\natural}_{\pi}(\{t_2\})$ | FN | FN | TN |
| $A^{\natural}_{\theta}(\{n_1\}) \cap B^{\natural}_{\pi}(\{n_2\})$ | FN | FN | TN |

**Table 3-8 - Fused *across I AND* classification system outcomes with respect to true partitions of the event set.**

There is something peculiar about the individual classification systems under the $\wedge^I$ fusion rule. The system will produce a FP if $A_\theta$ gives the $t_2$ element a $t_1$ label or it can produce a TN if it labels the $t_2$ element as a non-target. Similarly, classification system $B_\pi$ has a similar issue when classifying elements from the $t_1$ partition. If $B_\pi$ labels the $t_1$ event with a $t_2$ label, then the classification system produces a FP, but if it labels the same $t_1$ element with an n label, it produces a TN classification. Referring to the table above, this produces some interesting results for the fused system. It appears that if $A_\theta$ correctly identifies a type I target with the $t_1$ label and $B_\pi$ labels the same element as belonging to $n_2$, then the system should produce a true positive event. The reason this does not occur is that unless the true partition of the event sets is known, it is not known whether the element that system $B_\pi$ labeled was a true non-target or a $t_1$.

Consider the *across I OR* label fusion rule, which is denoted $\vee^I$. Note that table 3.7 remains true for the $\vee^I$ combined classification system. Consider the labels with respect to the true partitions of the event set and label sets:

| $\vee^I$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_1$ | $n_1 = \{n, t_2\}$ |
| $t_2$ | t | t |
| $n_2 = \{n, t_1\}$ | t | n |

Table 3-9 - Truth table for *across I OR* fused classification system.

| $\vee^{I}$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|
| $A^{\natural}_{\theta}(\{t_1\}) \cup B^{\natural}_{\pi}(\{t_2\})$ | TP | TP | FP |
| $A^{\natural}_{\theta}(\{t_1\}) \cup B^{\natural}_{\pi}(\{n_2\})$ | TP | TP | FP |
| $A^{\natural}_{\theta}(\{n_1\}) \cup B^{\natural}_{\pi}(\{t_2\})$ | TP | TP | FP |
| $A^{\natural}_{\theta}(\{n_1\}) \cup B^{\natural}_{\pi}(\{n_2\})$ | FN | FN | TN |

Table 3-10 - *Across I OR* fused classification system outcomes with respect to the true partitions of the event set.

There are a few events of interest in table 3.10. First, consider the instance where $A_\theta$ labels the element "$t_1$" and $B_\pi$ labels the element as "$n_2$" = $\{t_1, n\}$ and the combined classification system returns a TP with respect to $\Gamma_{t_2}$. Both individual classification systems in this case are wrong, but considering that $A_\theta$ labeled the element as a target (even though it is the wrong target label), the combined system is correct in labeling the element a target. The other unusual occurrence happens when $A_\theta$ labels the element with an $n_1$ label and $B_\pi$ labels the same element with a $t_2$ label. With respect to the partition $\Gamma_{t_1}$, both individual systems were incorrect, but because $B_\pi$ labeled the element with a target label, the combined classification system correctly labeled the element from $\Gamma_{t_1}$ as a target.

The following is a brief treatment of the outcomes that one would expect from fused *across II* and *across III* fused classification systems. Note that for *across* case 2 ($\wedge^{II} \backslash \vee^{II}$), the target partition that $A_\theta$ is tuned to classify is a subset of the target partition that classification system $B_\pi$ is designed to classify. Recall, also, that for *across* case 3 ($\wedge^{III} \backslash \vee^{III}$) that $A_\theta$ is designed to classify targets $t_1$ and $t_2$ and classification system $B_\pi$ is designed to classify targets $t_2$ and $t_3$, where $t_2$ forms a subset for both classification systems.

| $\wedge^{II}$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_1$ | $n_1 = \{n, t_2\}$ |
| $t = \{t_1, t_2\}$ | t | n |
| n | n | n |

Table 3-11 - *Across II AND* combined classification system truth table.

| $\vee^{II}$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_1$ | $n_1 = \{n, t_2\}$ |
| $t = \{t_1, t_2\}$ | t | t |
| n | t | n |

Table 3-12 - *Across II OR* combined classification system truth table.

| $\wedge^{II}$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|
| $A^{\natural}_\theta(\{t_1\}) \cap B^{\natural}_\pi(\{t\})$ | TP | TP | FP |
| $A^{\natural}_\theta(\{t_1\}) \cap B^{\natural}_\pi(\{n\})$ | FN | FN | TN |
| $A^{\natural}_\theta(\{n_1\}) \cap B^{\natural}_\pi(\{t\})$ | FN | FN | TN |
| $A^{\natural}_\theta(\{n_1\}) \cap B^{\natural}_\pi(\{n\})$ | FN | FN | TN |

Table 3-13 - *Across II AND* fused classification system outcomes with respect to the true partitions of the event set.

| $\vee^{II}$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|
| $A^{\natural}_\theta(\{t_1\}) \cup B^{\natural}_\pi(\{t\})$ | TP | TP | FP |
| $A^{\natural}_\theta(\{t_1\}) \cup B^{\natural}_\pi(\{n\})$ | TP | TP | FP |
| $A^{\natural}_\theta(\{n_1\}) \cup B^{\natural}_\pi(\{t\})$ | TP | TP | FP |
| $A^{\natural}_\theta(\{n_1\}) \cup B^{\natural}_\pi(\{n\})$ | FN | FN | TN |

Table 3-14- *Across II OR* fused classification system outcomes with respect to the true partitions of the event set.

| $\wedge^{III}$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_{12} = \{t_1, t_2\}$ | $n_1 = \{n, t_3\}$ |
| $t_{23} = \{t_2, t_3\}$ | t | n |
| $n_{23} = \{n, t_1\}$ | n | n |

Table 3-15 - *Across III AND* combined classification system truth table.

| $\vee^{III}$ | $A_\theta$ | |
|---|---|---|
| $B_\pi$ | $t_{12} = \{t_1, t_2\}$ | $n_1 = \{n, t_3\}$ |
| $t_{23} = \{t_2, t_3\}$ | t | t |
| $n_{23} = \{n, t_1\}$ | t | n |

Table 3-16 - *Across III OR* combined classification system truth table.

| $\wedge^{III}$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_{t_3}$ (True target type III partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|---|
| $A^{\natural}_{\theta}(\{t_{12}\}) \cap B^{\natural}_{\pi}(\{t_{23}\})$ | TP | TP | TP | FP |
| $A^{\natural}_{\theta}(\{t_{12}\}) \cap B^{\natural}_{\pi}(\{n_{23}\})$ | FN | FN | FN | TN |
| $A^{\natural}_{\theta}(\{n_{12}\}) \cap B^{\natural}_{\pi}(\{t_{23}\})$ | FN | FN | FN | TN |
| $A^{\natural}_{\theta}(\{n_{12}\}) \cap B^{\natural}_{\pi}(\{n_{23}\})$ | FN | FN | FN | TN |

**Table 3-17 - *Across III AND* fused classification system outcomes with respect to the true partitions of the event set.**

| $\vee^{III}$ | $\Gamma_{t_1}$ (True target type I partition) | $\Gamma_{t_2}$ (True target type II partition) | $\Gamma_{t_3}$ (True target type III partition) | $\Gamma_n$ (True non-target partition) |
|---|---|---|---|---|
| $A^{\natural}_{\theta}(\{t_{12}\}) \cup B^{\natural}_{\pi}(\{t_{23}\})$ | TP | TP | TP | FP |
| $A^{\natural}_{\theta}(\{t_{12}\}) \cup B^{\natural}_{\pi}(\{n_{23}\})$ | TP | TP | TP | FP |
| $A^{\natural}_{\theta}(\{n_{12}\}) \cup B^{\natural}_{\pi}(\{t_{23}\})$ | TP | TP | TP | FP |
| $A^{\natural}_{\theta}(\{n_{12}\}) \cup B^{\natural}_{\pi}(\{n_{23}\})$ | FN | FN | FN | TN |

**Table 3-17 - *Across III OR* fused classification system outcomes with respect to the true partitions of the event set.**

### *3.5 Probability Theory and its Applications to Classification Theory*

In order to create a ROC curve for a fused classification system, one must choose whether *within* or *across* fusion is applicable based on setting and then apply the Boolean rule of choice. Once these have been established, there exists a way to develop a probability expression that models the performance of the classification system, namely, the ROC curve. The background theory presented here provides a way to derive the ROC curve expression for *within* and *across* label fusion using only the performances of the individual classification systems. It is assumed that classification systems $A_{\theta}$ and $B_{\pi}$ are

conditionally independent with respect to the non-target event set. Independence and conditional independence of events are now defined.

**Definition 3.11** *Let ($\Gamma$, G, $P_\Gamma$) define a probability measure space. Let C and D be any sets in G. Then the following are true:*

$$P(C \cap D^C) = P(C) - P(D \cap C)$$

$$P(D \cup C) = P(D) + P(C) - P(D \cap C)$$

**Definition 3.12** *(Independence of Events) Let ($\Gamma$, G, $P_\Gamma$) be a probability measure space. The collection of events $\{E_1, E_2, \ldots, E_n\} \subset G$ is said to be independent if*

$$P\left(\bigcap_{i=1}^{n} E_i\right) = \prod_{i=1}^{n} P(E_i)$$

**Definition 3.13** *(Conditional Independence of Events) Let ($\Gamma$, G, $P_\Gamma$) be a probability measure space. The collection of events $\{E_1, E_2, \ldots, E_n\} \subset G$ are said to be conditionally independent with respect to event $E_j$ if for any sub-collection* $\left\{E_{i_1}, E_{i_2}, \ldots, E_{i_j}\right\} \subset \{E_1, E_2, \ldots, E_n\}$ *then*

$$P\left(\bigcap_{i=1}^{m} [E_i | E_j]\right) = \prod_{i=1}^{m} P(E_i | E_j)$$

*3.6 Correlation*

In (*Schubert et al: 2005)*, the functions for the ROC curves associated with the different types of label fusion were derived under the assumption that classification systems were correlated. Using the results of (*Schubert et al: 2005) correlation* between two classification systems takes the functional form below.

$$\rho[A_\theta, B_\pi] = \frac{[C_{A\wedge B} - C_A C_B]}{\sqrt{C_A(1-C_A)}\sqrt{C_B(1-C_B)}} \qquad (3.10)$$

Where

$$C_A = P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta) \qquad (3.11)$$

$$C_B = P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)$$

$$C_{A\wedge B} = P(\Gamma_t)P_{TP}(A_\theta \wedge B_\pi) + P(\Gamma_n)P_{FP}(A_\theta \wedge B_\pi)$$

Notice that $C_A$, $C_B$, $C_{A \wedge B}$, called the *cost functions*, are weighted functions involving the false positive and true positive rates that constitute the ROC curves of classification systems $A_\theta$ and $B_\pi$. They arise naturally from the expressions for the *expected value* of classification systems $A_\theta$ and $B_\pi$. When the two classification systems are independent, the correlation term goes to zero, i.e., when $C_{A\wedge B} = C_A C_B$.

Next, the functions $T_A(p)$ and $T_B(q)$ are defined.

**Def 3.16 –** *(T functions) – The T functions are defined to be the cost functions for classification systems $A_\theta$ and $B_\pi$ that are maximized over their respective parameter sets. These functions are the optimal performance points on the ROC curves of classification systems $A_\theta$ and $B_\pi$, respectively.*

$$\begin{aligned} T_A(p) &= \max_{\theta \in \Theta} C_{A_\theta} \\ &= \max_{\theta \in \Theta}[P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)] \\ &= \max_{\theta \in \Theta}[P(\Gamma_t)f_A(p) + P(\Gamma_n)p] \end{aligned}$$

$$T_B(q) = \max_{\pi \in \Pi} C_{B_\pi}$$

$$= \max_{\pi \in \Pi}[P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)]$$

$$= \max_{\pi \in \Pi}[P(\Gamma_t)f_B(q) + P(\Gamma_n)q]$$

These functions maximize over the parameters permitting the use of values from the ROC curve for classification systems $A_\theta$ and $B_\pi$. These functions take the same general form for any type of label fusion, however the prior probability weighting on the TP and FP values changes depending on the type of *across* or *within* label fusion being applied.

It is further derived in (*Schubert et al: 2005)* that the ROC curve for any *AND* label-fused system where the correlation remains constant for any combination of parameters θ and π takes the following form.

$$f_{C^{AND}}(r) = \frac{1}{P(\Gamma_t)} \max_{pq=r} g_p\big(T_A(p), T_B(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r \qquad (3.12)$$

The value “r” is the product of FP values *p* and *q* from the ROC curves of individual classification systems $A_\theta$ and $B_\pi$ via the conditional independence assumptions on the non-target partition of the event set. The function $g_\rho$ takes the following form.

$$g_\rho = \rho\sqrt{T_A(p)\big(1 - T_A(p)\big)}\sqrt{T_B(q)\big(1 - T_B(q)\big)} + T_A(p)T_B(q) \qquad (3.13)$$

In a similar fashion, the ROC curve for any *OR* label-fused system under the assumption of constant correlation between the classification systems at each parameter combination may be defined.

$$f_{C^{OR}}(r) = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} h_p\big(T_A(p), T_B(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r \qquad (3.14)$$

For the combined *OR* ROC curve, the quantity "r" takes the form of the *OR* probability statement for the false positive value of classification system $A_\theta$ or $B_\pi$ and $h_\rho$ is defined as the following.

$$\begin{aligned} h_\rho = {} & T_A(p) + T_B(q) \\ & - \rho\sqrt{T_A(p)\big(1 - T_A(p)\big)}\sqrt{T_B(q)\big(1 - T_B(q)\big)} \\ & - T_A(p)T_B(q) \end{aligned} \qquad (3.15)$$

For a more in depth discussion of the derivation of these terms, the reader is directed to (*Schubert et al: 2005).*

### *3.7 ROC Curves for Different Types of Label Fusion*

#### *3.7.1 Within Label-Fused ROC Curves*

Now that a general equation for a label-fused ROC curve has been developed, it can be applied to the different cases of label fusion. First, consider the *within AND*

combined classification system. As elaborated in *(Schubert: 2005)* the combined *AND* classification system under constant correlation takes the following form.

**Def 3.17 –** *Within AND ROC curve – Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination $\theta$ and $\pi$ the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system $A_\theta$ AND $B_\pi$.*

$$P_{TP}^{W}(A_\theta \wedge B_\pi) = \frac{1}{P(\Gamma_t)} \max_{pq=r} g_\rho\left(T_A^W(p), T_B^W(q)\right) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$\begin{aligned} P_{FP}^{W}(A_\theta \wedge B_\pi) &= P_{FP}(A_\theta) P_{FP}(B_\pi) \\ &= pq \end{aligned}$$

*Where*

$$T_A^W(p) = \max_{\theta \in \Theta} [P(\Gamma_t) f_A(p) + P(\Gamma_n) p]$$

$$T_B^W(q) = \max_{\pi \in \Pi} [P(\Gamma_t) f_B(q) + P(\Gamma_n) P_{FP}(B_\pi)]$$

$$g_\rho\left(T_A^W(p), T_B^W(q)\right) = \rho \left( \sqrt{T_A^W(p)\left(1 - T_A^W(p)\right)} \sqrt{T_B^W(q)\left(1 - T_B^W(q)\right)} \right) + T_A^W(p) T_B^W(q)$$

Clearly, if the two systems are uncorrelated, then the correlation constant, ρ, is zero and the calculation of the TP value becomes the product of the T functions for classification systems $A_\theta$ and $B_\pi$ minus the weighted FP. Correlation, by definition, is bounded on the interval [-1, 1] where ρ = -1 implies the two classification systems are inversely correlated and ρ = 1 implies the two classification systems are directly correlated.

The *within OR* combined classification system is defined in a similar fashion.

**Def 3.18** – *Within OR ROC curve - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination $\theta$ and $\pi$, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{W}(A_\theta \vee B_\pi) = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} h_\rho\big(T_A^W(p), T_B^W(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$P_{FP}^{W}(A_\theta \vee B_\pi) = P_{FP}(A_\theta) + P_{FP}(B_\pi) - P_{FP}(A_\theta)P_{FP}(B_\pi)$$

$$= p + q - pq$$

*Where*

$$T_A^W(p) = \max_{\theta\in\Theta}[P(\Gamma_t)f_A(p) + P(\Gamma_n)p]$$

$$T_B^W(q) = \max_{\pi\in\Pi}[P(\Gamma_t)f_B(q) + P(\Gamma_n)q]$$

$$h_\rho\big(T_A^W(p), T_B^W(q)\big)$$

$$= T_A^W(p) + T_B^W(q) - \rho\left(\sqrt{T_A^W(p)\big(1 - T_A^W(p)\big)}\sqrt{T_B^W(q)\big(1 - T_B^W(q)\big)}\right)$$

$$- T_A^W(p)T_B^W(q)$$

If the two systems are uncorrelated, the $h_\rho$ function simplifies to the form of the independent *OR* probability calculation using the T functions.

*3.7.2 Across I Label-Fused ROC Curves*

The T functions for the *across I* label-fused ROC curves closely resemble those of the *within* label-fused ROC curves. The derivation of these functions can be found in (*Schubert: 2005*). ***The major difference between across and within label-fused ROC curves is the prior probability weighting on individual classification systems $A_\theta$ and $B_\pi$.***

**Def 3.19 –** *Across I AND ROC curve - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination $\theta$ and $\pi$, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{I}(A_\theta \wedge B_\pi) = \frac{1}{P(\Gamma_t)} \max_{pq=r} g_\rho\big(T_A^I(p), T_B^I(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$P_{FP}^{I}(A_\theta \wedge B_\pi) = P_{FP}(A_\theta) P_{FP}(B_\pi)$$

$$= pq$$

*Where*

$$T_A^I(p) = \max_{\theta \in \Theta} \left[P\big(\Gamma_{t_1}\big) f_A(p) + P\big(\Gamma_{n_1}\big) p\right]$$

$$T_B^I(q) = \max_{\pi \in \Pi} \left[P\big(\Gamma_{t_2}\big) f_B(q) + P\big(\Gamma_{n_2}\big) q\right]$$

$$g_\rho\big(T_A^I(p), T_B^I(q)\big) = \rho\left(\sqrt{T_A^I(p)\big(1 - T_A^I(p)\big)}\sqrt{T_B^I(q)\big(1 - T_B^I(q)\big)}\right) + T_A^I(p) T_B^I(q)$$

Note that the prior probability associated with the T functions has changed and is dependent upon choice of label fusion rule.

Next, the formula for the *across I OR* combined ROC curve is defined.

**Def 3.20 –** *Across I OR ROC curve - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination $\theta$ and $\pi$, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{I}(A_\theta \vee B_\pi) = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} h_\rho\big(T_A^I(p), T_B^I(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$P_{FP}^{I}(A_\theta \vee B_\pi) = P_{FP}(A_\theta) + P_{FP}(B_\pi) - P_{FP}(A_\theta) P_{FP}(B_\pi)$$

$$= p + q - pq$$

*Where*

$$T_A^I(p) = \max_{\theta \in \Theta}\left[P\left(\Gamma_{t_1}\right)f_A(p) + P\left(\Gamma_{n_1}\right)p\right]$$

$$T_B^I(q) = \max_{\pi \in \Pi}\left[P\left(\Gamma_{t_2}\right)f_B(q) + P\left(\Gamma_{n_2}\right)q\right]$$

$$h_\rho\left(T_A^I(p), T_B^I(q)\right)$$

$$= T_A^I(p) + T_B^I(q) - \rho\left(\sqrt{T_A^I(p)\left(1 - T_A^I(p)\right)}\sqrt{T_B^I(q)\left(1 - T_B^I(q)\right)}\right)$$

$$- T_A^I(p)T_B^I(q)$$

The prior probability has a significant effect on combined ROC curves for the *across I* combined classification system. Performance is greatly impacted under the *AND* rule as there are few instances where the combined classification system should agree about the target label for a given element.

*3.7.3 – Across II Label-Fused ROC Curves*

In the section below, the formulas for the *AND* and the *OR* ROC curves are derived for the *across II* combined classification system.

**Def 3.21 –** *Across II AND ROC curve - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination $\theta$ and $\pi$, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{II}(A_\theta \wedge B_\pi) = \frac{1}{P(\Gamma_t)}\max_{pq=r} g_\rho\left(T_A^{II}(p), T_B^{II}(q)\right) - \frac{P(\Gamma_n)}{P(\Gamma_t)}r$$

$$P_{FP}^{II}(A_\theta \wedge B_\pi) = P_{FP}(A_\theta)P_{FP}(B_\pi)$$

$$= pq$$

*Where*

$$T_A^{II}(p) = \max_{\theta \in \Theta}\left[P\left(\Gamma_{t_1}\right)f_A(p) + P\left(\Gamma_{n_1}\right)p\right]$$

$$T_B^{II}(q) = \max_{\pi \in \Pi}[P(\Gamma_t)f_B(q) + P(\Gamma_n)q]$$

$$g_\rho\big(T_A^{II}(p), T_B^{II}(q)\big) = \rho\left(\sqrt{T_A^{II}(p)\big(1 - T_A^{II}(p)\big)}\sqrt{T_B^{II}(q)\big(1 - T_B^{II}(q)\big)}\right) + T_A^{II}(p)T_B^{II}(q)$$

Note that the prior probabilities associated with the T functions of system $B_\pi$ under *across II* label fusion are identical to the prior probabilities associated with system $B_\pi$ under *within* label fusion. Hence, *across II* fusion becomes *within* fusion when the prior probabilities associated with system $A_\theta$ are equivalent to the prior probabilities associated with system $B_\pi$.

Next, the *across II OR* ROC curve formula is defined

**Def 3.22 –** *Across II OR ROC curve* - *Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination θ and π, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{II}(A_\theta \vee B_\pi) = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} h_\rho\big(T_A^{II}(p), T_B^{II}(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)}r$$

$$\begin{aligned} P_{FP}^{II}(A_\theta \vee B_\pi) &= P_{FP}(A_\theta) + P_{FP}(B_\pi) - P_{FP}(A_\theta)P_{FP}(B_\pi) \\ &= p + q - pq \end{aligned}$$

*Where*

$$T_A^{II}(p) = \max_{\theta \in \Theta}\big[P\big(\Gamma_{t_1}\big)f_A(p) + P\big(\Gamma_{n_1}\big)p\big]$$

$$T_B^{II}(q) = \max_{\pi \in \Pi}[P(\Gamma_t)f_B(q) + P(\Gamma_n)q]$$

$$\begin{aligned} &h_\rho\big(T_A^{II}(p), T_B^{II}(q)\big) \\ &\quad = T_A^{II}(p) + T_B^{II}(q) - \rho\left(\sqrt{T_A^{II}(p)\big(1 - T_A^{II}(p)\big)}\sqrt{T_B^{II}(q)\big(1 - T_B^{II}(q)\big)}\right) \\ &\quad - T_A^{II}(p)T_B^{II}(q) \end{aligned}$$

*3.7.4 Across III Label-Fused ROC Curves*

Below, the ROC curves for *across III* label-fused systems are defined.

**Def 3.23 –** *Across III AND function - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination θ and π, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{III}(A_\theta \wedge B_\pi) = \frac{1}{P(\Gamma_t)} \max_{pq=r} g_\rho\big(T_A^{III}(p), T_B^{III}(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$P_{FP}^{III}(A_\theta \wedge B_\pi) = P_{FP}(A_\theta)P_{FP}(B_\pi)$$

$$= pq$$

*Where*

$$T_A^{III}(p) = \max_{\theta \in \Theta}\left[P\big(\Gamma_{t_{12}}\big)f_A(p) + P\big(\Gamma_{n_{12}}\big)p\right]$$

$$T_B^{III}(q) = \max_{\pi \in \Pi}\left[P\big(\Gamma_{t_{23}}\big)f_B(q) + P\big(\Gamma_{n_{23}}\big)q\right]$$

$$g_\rho\big(T_A^{III}(p), T_B^{III}(q)\big)$$

$$= \rho\left(\sqrt{T_A^{III}(p)\big(1 - T_A^{III}(p)\big)}\sqrt{T_B^{III}(q)\big(1 - T_B^{III}(q)\big)}\right) + T_A^{III}(p)T_B^{III}(q)$$

**Def 3.24** – *Across III OR ROC curve - Assume that the ROC curves for systems $A_\theta$ and $B_\pi$ are known. Then for a given parameter combination θ and π, the (FP, TP) ordered pair constitutes a point on the ROC curve for the combined classification system.*

$$P_{TP}^{III}(A_\theta \vee B_\pi) = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} h_\rho\big(T_A^{III}(p), T_B^{III}(q)\big) - \frac{P(\Gamma_n)}{P(\Gamma_t)} r$$

$$P_{FP}^{III}(A_\theta \vee B_\pi) = P_{FP}(A_\theta) + P_{FP}(B_\pi) - P_{FP}(A_\theta)P_{FP}(B_\pi)$$

$$= p + q - pq$$

*Where*

$$T_A^{III}(p) = \max_{\theta \in \Theta}\left[P\left(\Gamma_{t_{12}}\right)f_A(p) + P\left(\Gamma_{n_{12}}\right)p\right]$$

$$T_B^{III}(q) = \max_{\pi \in \Pi}\left[P\left(\Gamma_{t_{23}}\right)f_B(q) + P\left(\Gamma_{n_{23}}\right)q\right]$$

$$h_\rho\left(T_A^{III}(p), T_B^{III}(q)\right)$$

$$= T_A^{III}(p) + T_B^{III}(q) - \rho\left(\sqrt{T_A^{III}(p)\left(1 - T_A^{III}(p)\right)}\sqrt{T_B^{III}(q)\left(1 - T_B^{III}(q)\right)}\right)$$

$$- T_A^{III}(p)T_B^{III}(q)$$

In section IV, the results of the research will be discussed along with derivations for the functions that describe the bias between *across* and *within* label-fused classification systems. In section V, the simulation will be analyzed. This simulation compares the ROC curves of *within* and *across* combined classification systems, the bias that exists between these ROC curves, and different environmental factors that can affect the bias.

## *IV Results*

Recall that the aim of this research is to describe and quantify the bias between classification systems under the assumptions of *within* and *across* label fusion. The results of (*Schubert et al: 2005*) suggest that there is some inherent difference in the performance of classification systems under the different types of *across* fusion as compared to *within* fusion. First, the notion of bias must be developed. Generally, bias will be defined as the difference in true positive rate for a given, fixed false positive rate. Thus, bias may be viewed graphically when depicting the *within* label-fused ROC curve and the *across* label-fused ROC curve on the same plot and observing the vertical distance between the two curves for a fixed, x-axis (false positive) value. Label-fused ROC curve formulas expressed in terms of the performance of the individual classification systems presented previously may be used to create computational formulas for bias in terms of the individual systems. Thus, adjustments to fused system performance may be generated in a flexible manner. Formulas for bias between *within* label-fused ROC curves and each of the types of *across* label-fused ROC curves are derived below.

### *4.1 Bias Between Across I and Within ROC Curves*

The bias between *across I* and *within* label-fused ROC curves may be expressed in terms of the fusion rule and the ROC curves of the individual classification systems.

**Definition 4.1.1** (*Performance Bias: Across I versus Within, Boolean AND rule*) The performance bias between fused *across I* and *within* ROC curves under the Boolean *AND* rule is defined to be the following.

$$Bias_{D_{A\wedge B}}^{I\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ I\ AND}(r)$$

$$Bias_{D_{A\wedge B}}^{I\ vs\ W} = \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[g_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p'q'=r}\left[g_\rho\left(T_A^I(p'), T_B^I(q')\right)\right]\right)$$

It is possible that the values *p* and *q* whose product produced the combined false positive rate, r, may not be the same for both label-fused ROC curves. Hence, p may not be equal to p' and q may not be equal to q'. If the values p and q are the same for both label-fused ROC curves, then the formula can simplify slightly to the following:

$$Bias\ _{D_{A\wedge B}}^{I\ vs\ W} = \frac{1}{P(\Gamma_t)}\max_{pq=r}\left[T_A^W(p)T_B^W(q) - T_A^I(p)T_B^I(q)\right]$$

For the false positive values *p* and *q* to be the same for *across* and *within* classification systems, the combined false positive rate must be invariant with respect to the type of label fusion. That is to say, if the false positive values are calculated in a way that the fusion rule has no bearing on the computation of values *p* and *q* for individual classification systems, then this assumption can be met. For example, in the simulation of section V, the values for p and q are generated using parameter values and the normal CDF. Random sampling of values from the non-target distribution may lead to instances where p ≠ p' or q ≠ q'.

The following two theorems are concerned with deriving the formula for performance bias between the *across I* and *within* label-fused ROC curves under the Boolean *AND* rule. The first theorem makes no assumption about the equality of p, p', q, and q'. The second theorem is a simplification when $p = p'$ and $q = q'$.

**Theorem 4.1.1** *(Performance Bias: Across I versus Within label-fused ROC curves, Boolean AND rule) Let* $D_{A\wedge B}^I$ *be the ROC curve for the Boolean AND label-fused across I system. Let* $D_{A\wedge B}^W$ *be the ROC curve for the Boolean AND label-fused within system.*

*Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r = max p'q', the bias between across I and within label-fused ROC curves under the AND rule is*

$$Bias_{D_{A \wedge B}}^{I\ vs\ W} = \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}[T_A^W(p)T_B^W(q)] - \max_{p'q'=r}[T_A^I(p')T_B^I(q')]\right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation constant, ρ, is zero. Then, bias may be expressed as:

$$\begin{aligned}
Bias_{D_{A \wedge B}}^{I\ vs\ W} &= f_C^{Within\ AND}(r) - f_C^{Across\ I\ AND}(r) \\
&= \frac{1}{P(\Gamma_t)}\max_{pq=r}\left[g_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right] \\
&\quad - \frac{1}{P(\Gamma_t)}\max_{p'q'=r}\left[g_\rho\left(T_A^I(p'), T_B^I(q')\right) - P(\Gamma_n)r\right] \\
&= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}[T_A^W(p)T_B^W(q) - P(\Gamma_n)r] - \max_{p'q'=r}[T_A^I(p')T_B^I(q') - P(\Gamma_n)r]\right) \\
&= \frac{1}{P(\Gamma_t)}\Big(\max_{pq=r}\big[\left(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\right)\big(P(\Gamma_t)P_{TP}(B_\pi) \\
&\quad + P(\Gamma_n)P_{FP}(B_\pi)\big)\big] \\
&\quad - \max_{p'q'=r}\Big[\left(P\left(\Gamma_{t_1}\right)P_{TP}(A_\theta) + P\left(\Gamma_{n_1}\right)P_{FP}(A_\theta)\right)\Big(P\left(\Gamma_{t_2}\right)P_{TP}(B_\pi) \\
&\quad + P\left(\Gamma_{n_2}\right)P_{FP}(B_\pi)\Big)\Big]\Big) \\
&= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}[T_A^W(p)T_B^W(q)] + \max_{p'q'=r}[T_A^I(p')T_B^I(q')]\right)
\end{aligned}$$

■

**Theorem 4.1.2** *(Performance Bias: Across I versus Within label-fused ROC curves, Boolean AND rule where p=p' and q=q') Let $D_{A \wedge B}^I$ be the ROC curve for the Boolean*

*AND label-fused across I system. Let $D_{A\wedge B}^{W}$ be the ROC curve for the Boolean AND label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r, the bias between across I and within label-fused ROC curves under the AND rule is*

$$Bias_{D_{A\wedge B}}^{I\,vs\,W} = \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}[T_A^W(p)T_B^W(q) - T_A^I(p)T_B^I(q)]\right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Assume the probability values *p* and *q* are invariant with respect to choice of label fusion rule. Then, bias may be expressed as:

$$\begin{aligned}
Bias_{D_{A\wedge B}}^{I\,vs\,W} &= f_C^{Within\,AND}(r) - f_C^{Across\,I\,AND}(r) \\
&= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[g_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right]\right. \\
&\quad \left. - \max_{p'q'=r}\left[g_\rho\left(T_A^I(p'), T_B^I(q')\right) - P(\Gamma_n)r\right]\right) \\
&= \frac{1}{P(\Gamma_t)}\max_{pq=r}\Big[\left(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\right)\left(P(\Gamma_t)P_{TP}(B_\pi)\right. \\
&\quad \left. + P(\Gamma_n)P_{FP}(B_\pi)\right) \\
&\quad - P(\Gamma_n)r - \left(\left(P\left(\Gamma_{t_1}\right)P_{TP}(A_\theta) + P\left(\Gamma_{n_1}\right)P_{FP}(A_\theta)\right)\left(P\left(\Gamma_{t_2}\right)P_{TP}(B_\pi)\right.\right. \\
&\quad \left.\left. + P\left(\Gamma_{n_2}\right)P_{FP}(B_\pi)\right)\right) + P(\Gamma_n)r\Big] \\
&= \frac{1}{P(\Gamma_t)}\max_{pq=r}\Big[P_{TP}(A_\theta)P_{TP}(B_\pi)\left(P(\Gamma_t)^2 - P\left(\Gamma_{t_1}\right)P\left(\Gamma_{t_2}\right)\right) \\
&\quad + P_{TP}(A_\theta)P_{FP}(B_\pi)\left(P(\Gamma_t)P(\Gamma_n) - P\left(\Gamma_{t_1}\right)P\left(\Gamma_{n_2}\right)\right) \\
&\quad + P_{FP}(A_\theta)P_{TP}(B_\pi)\left(P(\Gamma_t)P(\Gamma_n) - P\left(\Gamma_{n_1}\right)P\left(\Gamma_{t_2}\right)\right) \\
&\quad + P_{FP}(A_\theta)P_{FP}(B_\pi)\left(P(\Gamma_n)^2 - P\left(\Gamma_{n_1}\right)P\left(\Gamma_{n_2}\right)\right)\Big]
\end{aligned}$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} [T_A^W(p) T_B^W(q) - T_A^I(p) T_B^I(q)]$$

■

In a similar fashion to the Boolean *AND* rule, the bias formula for the Boolean *OR* rule will be defined and then the corresponding theorems for the two different cases regarding the false positive values will be derived.

**Definition 4.1.3** *(Performance Bias: Across I versus Within Label-Fused ROC curves, Boolean OR rule)* Bias between the Boolean OR label-fused *within* and *across* ROC curves is

$$Bias_{D_{A \vee B}}^{I\,vs\,W} = f_C^{Within\ OR}(r) - f_C^{Across\ I\ OR}(r)$$

$$= \frac{1}{P(\Gamma_t)} \left( \max_{p+q-pq=r} \left[ h_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p'+q'-p'q'=r} \left[ h_\rho\left(T_A^I(p'), T_B^I(q')\right)\right] \right)$$

**Theorem 4.1.3** (*Performance Bias: Across I versus Within Label-Fused ROC curves, Boolean OR rule) Let $D_{A \vee B}^I$ be the ROC curve for the Boolean OR label-fused across I system. Let $D_{A \vee B}^W$ be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max p+q-pq = r =max p'+q'-p'q', the bias between across I and within label-fused classification systems is*

$$Bias_{D_{A \vee B}}^{I\,vs\,W} = \frac{1}{P(\Gamma_t)} \left( \max_{p+q-pq=r} \left[ h_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p'+q'-p'q'=r} \left[ h_\rho\left(T_A^I(p'), T_B^I(q')\right)\right] \right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Then, bias may be expressed as:

$$Bias_{D_{A\vee B}}^{I\,vs\,W} = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right]$$

$$- \frac{1}{P(\Gamma_t)} \max_{p'+q'-p'q'=r}\left[h_\rho\left(T_A^I(p'), T_B^I(q')\right) - P(\Gamma_n)r\right]$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}[T_A^W(p)T_B^W(q) - P(\Gamma_n)r]$$

$$- \max_{p'+q'-p'q'=r}[T_A^I(p')T_B^I(q') - P(\Gamma_n)r]\Big)$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}[T_A^W(p) + T_B^W(q) - T_A^W(p)T_B^W(q) - P(\Gamma_n)r]$$

$$- \max_{p'+q'-p'q'=r}[T_A^I(p') + T_B^I(q') - T_A^I(p')T_B^I(q') - P(\Gamma_n)r]\Big)$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\Big[\big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)$$

$$+ \big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)$$

$$- \big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)\big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)\Big]$$

$$- \max_{p'+q'-p'q'=r}\Big[\Big(\Big(P\big(\Gamma_{t_1}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big)P_{FP}(A_\theta)\Big)$$

$$+ \Big(P\big(\Gamma_{t_2}\big)P_{TP}(B_\pi) + P\big(\Gamma_{n_2}\big)P_{FP}(B_\pi)\Big)$$

$$- \Big(P\big(\Gamma_{t_1}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big)P_{FP}(A_\theta)\Big)\Big(P\big(\Gamma_{t_2}\big)P_{TP}(B_\pi)$$

$$+ P\big(\Gamma_{n_2}\big)P_{FP}(B_\pi)\Big)\Big)\Big]\Big)$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p'+q'-p'q'=r}\left[h_\rho\left(T_A^I(p'), T_B^I(q')\right)\right]\Big)$$

■

Seeing as there is no simplification of the formula beyond attempting to group cross terms, it was decided to leave the result in the form of the h function for ease of interpretation.

**Theorem 4.1.4** *(Performance Bias: Across I versus Within Label-Fused ROC curves, Boolean OR rule where p=p' and q=q') Let* $D^I_{A \vee B}$ *be the ROC curve for the Boolean OR label-fused across I system. Let* $D^W_{A \vee B}$ *be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max p+q-pq = r, the bias between across I and within label-fused ROC curves is*

$$Bias^{I\,vs\,W}_{D_{A \vee B}} = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ h_\rho\left(T^W_A(p), T^W_B(q)\right) - h_\rho\left(T^I_A(p), T^I_B(q)\right) \right]$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Further, assume that *p* and *q* are invariant with respect to choice of label fusion rule. Then, the bias may be expressed as:

$$Bias^{\,I\,vs\,W}_{\,D_{A \vee B}} = f_C^{Within\,OR}(r) - f_C^{Across\,I\,OR}(r)$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[ h_\rho\left(T^W_A(p), T^W_B(q)\right) - P(\Gamma_n) r$$

$$- \left( h_\rho\left(T^I_A(p), T^I_B(q)\right) - P(\Gamma_n) r \right) \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[ \left( T^W_A(p) + T^W_B(q) - T^W_A(p) T^W_B(q) \right)$$

$$- \left( T^I_A(p) + T^I_B(q) - T^I_A(p) T^I_B(q) \right) \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[ \big( P(\Gamma_t) P_{TP}(A_\theta) + P(\Gamma_n) P_{FP}(A_\theta) \big)$$

$$+ \big( P(\Gamma_t) P_{TP}(B_\pi) + P(\Gamma_n) P_{FP}(B_\pi) \big)$$

$$- \big( P(\Gamma_t) P_{TP}(A_\theta) + P(\Gamma_n) P_{FP}(A_\theta) \big) \big( P(\Gamma_t) P_{TP}(B_\pi) + P(\Gamma_n) P_{FP}(B_\pi) \big)$$

$$- \Big( \Big( P\big(\Gamma_{t_1}\big) P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big) P_{FP}(A_\theta) \Big)$$

$$+ \Big( P\big(\Gamma_{t_2}\big) P_{TP}(B_\pi) + P\big(\Gamma_{n_2}\big) P_{FP}(B_\pi) \Big)$$

$$- \Big( P\big(\Gamma_{t_1}\big) P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big) P_{FP}(A_\theta) \Big) \Big( P\big(\Gamma_{t_2}\big) P_{TP}(B_\pi) + P\big(\Gamma_{n_2}\big) P_{FP}(B_\pi) \Big) \Big) \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ h_\rho\big(T_A^W(p), T_B^W(q)\big) - h_\rho\big(T_A^I(p), T_B^I(q)\big) \right]$$

■

*4.2 Bias Between Across II and Within ROC Curves*

The bias between *across II* and *within* label-fused ROC curves may be expressed in terms of the fusion rule and the ROC curves of the individual classification systems.

**Definition 4.2.1** (*Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean AND rule*) Bias between the Boolean *AND* label-fused *within* and *across II* ROC curves is defined to be the following

$$Bias_{D_{A \wedge B}}^{II\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ II\ AND}(r)$$

$$= \frac{1}{P(\Gamma_t)} \left( \max_{pq=r} \left[ g_\rho\big(T_A^W(p), T_B^W(q)\big) \right] - \max_{p'q'=r} \left[ g_\rho\big(T_A^{II}(p'), T_B^{II}(q')\big) \right] \right)$$

Recall from section III that classification system $B_\pi$ under the *across II* fusion rule is designed to classify all target types in the environment. As such, the system $B_\pi$ is the same for both the *within* and *across II* label-fused systems. That is to say

$$T_B^W(q) = T_B^{II}(q)$$

This permits a nice simplification of the formula when q = q'.

**Theorem 4.2.1** *(Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean AND rule) Let* $D^{II}_{A \wedge B}$ *be the ROC curve for the Boolean AND label-fused across II system. Let* $D^{W}_{A \wedge B}$ *be the ROC curve for the Boolean AND label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r = max p'q', the bias between across II and within label-fused ROC curves is*

$$Bias^{II\ vs\ W}_{D_{A \wedge B}} = \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[g_\rho\left(T^W_A(p), T^W_B(q)\right)\right] - \max_{p'q'=r}\left[g_\rho\left(T^{II}_A(p'), T^{II}_B(q')\right)\right]\right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Then, bias may be expressed as:

$$\begin{aligned}
Bias^{II\ vs\ W}_{D_{A \wedge B}} &= f_C^{Within\ AND}(r) - f_C^{Across\ II\ AND}(r) \\
&= \frac{1}{P(\Gamma_t)}\max_{pq=r}\left[g_\rho\left(T^W_A(p), T^W_B(q)\right) - P(\Gamma_n)r\right] \\
&\quad - \frac{1}{P(\Gamma_t)}\max_{p'q'=r}\left[g_\rho\left(T^{II}_A(p'), T^{II}_B(q')\right) - P(\Gamma_n)r\right] \\
&= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[T^W_A(p)T^W_B(q) - P(\Gamma_n)r\right] - \max_{p'q'=r}\left[T^{II}_A(p')T^{II}_B(q') - P(\Gamma_n)r\right]\right) \\
&= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[\left(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\right)\left(P(\Gamma_t)P_{TP}(B_\pi)\right.\right.\right. \\
&\quad \left. + P(\Gamma_n)P_{FP}(B_\pi)\right) - P(\Gamma_n)r\Big] \\
&\quad - \max_{p'q'=r}\left[\left(P\left(\Gamma_{t_1}\right)P_{TP}(A_\theta) + P\left(\Gamma_{n_1}\right)P_{FP}(A_\theta)\right)\left(P(\Gamma_t)P_{TP}(B_\pi)\right.\right. \\
&\quad \left.\left.\left. + P(\Gamma_n)P_{FP}(B_\pi)\right) - P(\Gamma_n)r\right]\right)
\end{aligned}$$

$$= \frac{1}{P(\Gamma_t)} \Big( \max_{pq=r} \big[ \big( P(\Gamma_t) P_{TP}(A_\theta) + P(\Gamma_n) P_{FP}(A_\theta) \big) \big( P(\Gamma_t) P_{TP}(B_\pi)$$

$$+ P(\Gamma_n) P_{FP}(B_\pi) \big) - 1 \big]$$

$$- \max_{p'q'=r} \Big[ \Big( P\big(\Gamma_{t_1}\big) P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big) P_{FP}(A_\theta) \Big) \big( P(\Gamma_t) P_{TP}(B_\pi)$$

$$+ P(\Gamma_n) P_{FP}(B_\pi) \big) \Big] \Big)$$

$$= \frac{1}{P(\Gamma_t)} \Big( \max_{pq=r} \big[ g_\rho \big( T_A^W(p), T_B^W(q) \big) \big] - \max_{p'q'=r} \big[ g_\rho \big( T_A^{II}(p'), T_B^{II}(q') \big) \big] \Big)$$

■

**Theorem 4.2.2** *(Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean AND rule where p=p' and q=q') Let $D_{A\wedge B}^{II}$ be the ROC curve for the Boolean AND label-fused across II system. Let $D_{A\wedge B}^{W}$ be the ROC curve for the Boolean AND label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r, the bias between across II and within label-fused ROC curves is*

$$Bias_{D_{A\wedge B}}^{II\ vs\ W} = \frac{1}{P(\Gamma_t)} \max_{pq=r} \big[ T_B^W(q) \big( T_A^W(p) - T_A^{II}(p) \big) \big]$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero and the values p and q are invariant with respect to label fusion rule. Then, bias may be expressed as:

$$Bias_{C_{A\wedge B}}^{II\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ II\ AND}(r)$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \big[ g_\rho \big( T_A^W(p), T_B^W(q) \big) - P(\Gamma_n) r \big]$$

$$- \frac{1}{P(\Gamma_t)} \max_{pq=r} \big[ g_\rho \big( T_A^{II}(p), T_B^{II}(q) \big) - P(\Gamma_n) r \big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \left[ g_\rho\left(T_A^W(p), T_B^W(q)\right) - g_\rho\left(T_A^{II}(p), T_B^{II}(q)\right) \right]$$

Substitute in $T_B^W$ where appropriate

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} [T_A^W(p) T_B^W(q) - T_A^{II}(p) T_B^{II}(q)]$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \left[ T_A^W(p) T_B^W(q) - T_A^{II}(p)\left(T_B^W(q)\right) \right]$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \left[ T_B^W(q)\left(T_A^W(p) - T_A^{II}(p)\right) \right]$$

■

The bias formulas for *across II* versus *within* combined label-fused systems under the Boolean *OR* rule will be defined analogously.

**Def 4.2.3** (*Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean OR rule*) Bias between the Boolean *OR* label-fused *within* and *across II* ROC curves is defined to be the following

$$Bias_{D_{A \vee B}}^{II\ vs\ W} = f_C^{Within\ OR}(r) - f_C^{Across\ II\ OR}(r)$$

$$= \frac{1}{P(\Gamma_t)} \left( \max_{p+q-pq=r} \left[ h_\rho\left(T_A^W(p), T_B^W(q)\right) \right] - \max_{p'+q'-p'q'=r} \left[ h_\rho\left(T_A^{II}(p'), T_B^{II}(q')\right) \right] \right)$$

**Theorem 4.2.3** *(Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean OR rule) Let $D_{A \vee B}^{II}$ be the ROC curve for the Boolean OR label-fused across II system. Let $D_{A \vee B}^{W}$ be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, the bias between across II and within label-fused ROC curves is*

$$Bias_{D_{A\vee B}}^{II\ vs\ W} = \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\big[h_\rho\big(T_A^W(p), T_B^W(q)\big)\big]$$

$$- \max_{p'+q'-p'q'=r}\big[h_\rho\big(T_A^{II}(p'), T_B^{II}(q')\big)\big]\Big)$$

**Proof**: Under the assumption of independent classification systems A and B, the correlation constant, ρ, is zero. Then, bias may be expressed as:

$$Bias_{D_{A\vee B}}^{II\ vs\ W} = f_C^{Within\ OR}(r) - f_C^{Across\ II\ OR}(r)$$

$$= \frac{1}{P(\Gamma_t)}\max_{p+q-pq=r}\big[h_\rho\big(T_A^W(p), T_B^W(q)\big) - P(\Gamma_n)r\big]$$

$$- \frac{1}{P(\Gamma_t)}\max_{p'+q'-p'q'=r}\big[h_\rho\big(T_A^{II}(p'), T_B^{II}(q')\big) - P(\Gamma_n)r\big]$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}[T_A^W(p) + T_B^W(q) - T_A^W(p)T_B^W(q) - P(\Gamma_n)r]$$

$$- \max_{p'+q'-p'q'=r}[T_A^{II}(p') + T_B^{II}(q') - T_A^{II}(p')T_B^{II}(q') - P(\Gamma_n)r]\Big)$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\big[\big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)$$

$$+ \big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)$$

$$- \big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)\big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)\big]$$

$$- \max_{p'+q'-p'q'=r}\Big[\Big(\Big(P\big(\Gamma_{t_1}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big)P_{FP}(A_\theta)\Big)$$

$$+ \big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)$$

$$- \Big(P\big(\Gamma_{t_1}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_1}\big)P_{FP}(A_\theta)\Big)\big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)\Big)\Big]\Big)$$

$$= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\big[h_\rho\big(T_A^W(p), T_B^W(q)\big)\big] - \max_{p'+q'-p'q'=r}\big[h_\rho\big(T_A^{II}(p'), T_B^{II}(q')\big)\big]\Big)$$

■

**Theorem 4.2.4** *(Performance Bias: Across II versus Within Label-Fused ROC curves, Boolean OR rule where p=p' and q=q') Let $D_{A \vee B}^{II}$ be the ROC curve for the Boolean OR label-fused across II system. Let $D_{A \vee B}^{W}$ be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value max p+q-pq = r, the bias between across II and within label-fused classification systems is*

$$Bias_{D_{A \vee B}}^{II\ vs\ W} = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ \left(1 - T_B^{w}(q)\right)\left(T_A^{W}(p) - T_A^{II}(p)\right) \right]$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Further, assume that the values p and q are invariant with respect to label fusion rule. Then, bias may be expressed as:

$$\begin{aligned}
Bias_{D_{A \vee B}}^{II\ vs\ W} &= f_C^{Within\ OR}(r) - f_C^{Across\ II\ OR}(r) \\
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ h_\rho\left(T_A^{W}(p), T_B^{W}(q)\right) - P(\Gamma_n) r \right] \\
&\quad - \frac{1}{P(\Gamma_t)} \max_{p'+q'-p'q'=r} \left[ h_\rho\left(T_A^{II}(p'), T_B^{II}(q')\right) - P(\Gamma_n) r \right] \\
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ T_A^{W}(p) + T_B^{W}(q) - T_A^{W}(p) T_B^{W}(q) \right. \\
&\quad \left. - \left(T_A^{II}(p) + T_B^{II}(q) - T_A^{II}(p) T_B^{II}(q)\right) \right] \\
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ T_A^{W}(p) + T_B^{W}(q) - T_A^{W}(p) T_B^{W}(q) \right. \\
&\quad \left. - \left(T_A^{II}(p) + \left(T_B^{W}(q)\right) - T_A^{II}(p)\left(T_B^{W}(q)\right)\right) \right] \\
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} [T_A^{W}(p) - T_A^{W}(p) T_B^{w}(q) - T_A^{II}(p) + T_A^{II}(p) T_B^{W}(q)]
\end{aligned}$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ \left( T_A^W(p) - T_A^{II}(p) \right) - T_B^W(q) \left( T_A^W(p) - T_A^{II}(p) \right) \right]$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ \left( 1 - T_B^W(q) \right) \left( T_A^W(p) - T_A^{II}(p) \right) \right]$$

■

*4.3 Bias Between Across III and Within ROC Curves*

The bias between *across III* and *within* label-fused ROC curves may be expressed in terms of the fusion rule and the ROC curves of the individual classification systems.

**Definition 4.3.1** (*Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean AND rule*) Bias between the Boolean *AND* label-fused *within* and *across III* ROC curves is defined to be the following

$$Bias_{D_{A \wedge B}}^{III\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ III\ AND}(r)$$

$$= \frac{1}{P(\Gamma_t)} \left( \max_{pq=r} \left[ g_\rho \left( T_A^W(p), T_B^W(q) \right) \right] - \max_{p'q'=r} \left[ g_\rho \left( T_A^{III}(p'), T_B^{III}(q') \right) \right] \right)$$

**Theorem 4.3.1** *(Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean AND rule) Let $D_{A \wedge B}^{III}$ be the ROC curve for the Boolean AND label-fused across III system. Let $D_{A \wedge B}^{W}$ be the ROC curve for the Boolean AND label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r = max p'q', the bias between across III and within label-fused ROC curves is*

$$Bias_{D_{A \wedge B}}^{III\ vs\ W} = \frac{1}{P(\Gamma_t)} \left( \max_{pq=r} \left[ g_\rho \left( T_A^W(p), T_B^W(q) \right) \right] - \max_{p'q'=r} \left[ g_\rho \left( T_A^{III}(p'), T_B^{III}(q') \right) \right] \right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Then, bias may be expressed as:

$$Bias_{D_{A\wedge B}}^{III\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ III\ AND}(r)$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r}\left[g_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right]$$

$$- \frac{1}{P(\Gamma_t)} \max_{p'q'=r}\left[g_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right) - P(\Gamma_n)r\right]$$

$$= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}[T_A^W(p)T_B^W(q) - P(\Gamma_n)r] - \max_{p'q'=r}[T_A^{III}(p')T_B^{III}(q') - P(\Gamma_n)r]\right)$$

$$= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[\left(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\right)\left(P(\Gamma_t)P_{TP}(B_\pi)\right.\right.\right.$$

$$\left.\left. + P(\Gamma_n)P_{FP}(B_\pi)\right)\right] - \max_{p'q'=r}\left[\left(P\left(\Gamma_{t_{12}}\right)P_{TP}(A_\theta)\right.\right.$$

$$\left.\left.\left. + P\left(\Gamma_{n_{12}}\right)P_{FP}(A_\theta)\right)\left(P\left(\Gamma_{t_{23}}\right)P_{TP}(B_\pi) + P\left(\Gamma_{n_{23}}\right)P_{FP}(B_\pi)\right)\right]\right)$$

$$= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[\left(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\right)\left(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\right)\right]\right.$$

$$- \max_{p'q'=r}\left[\left(P\left(\Gamma_{t_1}\right)P_{TP}(A_\theta) + P\left(\Gamma_{n_1}\right)P_{FP}(A_\theta)\right)\left(P\left(\Gamma_{t_{23}}\right)P_{TP}(B_\pi)\right.\right.$$

$$\left.\left.\left. + P\left(\Gamma_{n_{23}}\right)P_{FP}(B_\pi)\right)\right]\right)$$

$$= \frac{1}{P(\Gamma_t)}\left(\max_{pq=r}\left[g_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p'q'=r}\left[g_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right)\right]\right)$$

■

**Theorem 4.2.2** *(Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean AND rule where p=p' and q=q') Let $D_{A\wedge B}^{III}$ be the ROC curve for the Boolean AND label-fused across III system. Let $D_{A\wedge B}^{W}$ be the ROC curve for the Boolean AND label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max pq = r, the bias between across III and within label-fused ROC curves is*

$$Bias_{D_{A\wedge B}}^{III\ vs\ W} = \frac{1}{P(\Gamma_t)} \max_{pq=r} [T_A^W(p) T_B^W(q) - T_A^{III}(p) T_B^{III}(q)]$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Further, assume that the values p and q are invariant with respect to choice of label fusion rule. Then, bias may be expressed as:

$$Bias_{D_{A\wedge B}}^{III\ vs\ W} = f_C^{Within\ AND}(r) - f_C^{Across\ III\ AND}(r)$$

$$= \frac{1}{P(\Gamma_t)} \left( \max_{pq=r} \left[ g_\rho \left( T_A^W(p), T_B^W(q) \right) - P(\Gamma_n) r \right] \right.$$

$$\left. - \max_{p'q'=r} \left[ g_\rho \left( T_A^{III}(p'), T_B^{III}(q') \right) - P(\Gamma_n) r \right] \right)$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \Big[ \big( P(\Gamma_t) P_{TP}(A_\theta) + P(\Gamma_n) P_{FP}(A_\theta) \big) \big( P(\Gamma_t) P_{TP}(B_\pi)$$

$$+ P(\Gamma_n) P_{FP}(B_\pi) \big)$$

$$- P(\Gamma_n) r - \Big( \Big( P\big(\Gamma_{t_{12}}\big) P_{TP}(A_\theta) + P\big(\Gamma_{n_{12}}\big) P_{FP}(A_\theta) \Big) \Big( P\big(\Gamma_{t_{23}}\big) P_{TP}(B_\pi)$$

$$+ P\big(\Gamma_{n_{23}}\big) P_{FP}(B_\pi) \Big) \Big) + P(\Gamma_n) r \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} \Big[ P_{TP}(A_\theta) P_{TP}(B_\pi) \Big( P(\Gamma_t)^2 - P\big(\Gamma_{t_{12}}\big) P\big(\Gamma_{t_{23}}\big) \Big)$$

$$+ P_{TP}(A_\theta) P_{FP}(B_\pi) \Big( P(\Gamma_t) P(\Gamma_n) - P\big(\Gamma_{t_{12}}\big) P\big(\Gamma_{n_{23}}\big) \Big)$$

$$+ P_{FP}(A_\theta) P_{TP}(B_\pi) \Big( P(\Gamma_t) P(\Gamma_n) - P\big(\Gamma_{n_{12}}\big) P\big(\Gamma_{t_{23}}\big) \Big)$$

$$+ P_{FP}(A_\theta) P_{FP}(B_\pi) \Big( P(\Gamma_n)^2 - P\big(\Gamma_{n_{12}}\big) P\big(\Gamma_{n_{23}}\big) \Big) \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{pq=r} [T_A^W(p) T_B^W(q) - T_A^{III}(p) T_B^{III}(q)]$$

■

The bias formulas for *across III* versus *within* combined label-fused systems under the Boolean *OR* rule will be defined analogously.

**Def 4.2.3** (*Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean OR rule*) Bias between the Boolean *OR* label-fused *within* and *across III* ROC curves is defined to be the following

$$Bias_{D_{A\vee B}}^{III\ vs\ W} = f_C^{Within\ OR}(r) - f_C^{Across\ II\ OR}(r)$$

$$= \frac{1}{P(\Gamma_t)}\left(\max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right] - \max_{p\prime+q\prime-p\prime q\prime=r}\left[h_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right) - P(\Gamma_n)r\right]\right)$$

**Theorem 4.2.3** *(Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean OR rule) Let* $D_{A\vee B}^{III}$ *be the ROC curve for the Boolean OR label-fused across III system. Let* $D_{A\vee B}^{W}$ *be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value, max p+q-pq = r = max p'+q'-p'q', the bias between across III and within label-fused ROC curves is*

$$Bias_{D_{A\vee B}}^{III\ vs\ W} = \frac{1}{P(\Gamma_t)}\left(\max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right)\right] - \max_{p\prime+q\prime-p\prime q\prime=r}\left[h_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right)\right]\right)$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Then, bias may be expressed as:

$$
\begin{aligned}
Bias\,{}_{D_{A\vee B}}^{III\ vs\ W} &= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n)r\right] \\
&\quad - \frac{1}{P(\Gamma_t)} \max_{p'+q'-p'q'=r}\left[h_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right) - P(\Gamma_n)r\right] \\
&= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}[T_A^W(p)T_B^W(q) - P(\Gamma_n)r] \\
&\quad - \max_{p'+q'-p'q'=r}[T_A^{III}(p')T_B^{III}(q') - P(\Gamma_n)r]\Big) \\
&= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}[T_A^W(p) + T_B^W(q) - T_A^W(p)T_B^W(q) - P(\Gamma_n)r] \\
&\quad - \max_{p'+q'-p'q'=r}[T_A^{III}(p') + T_B^{III}(q') - T_A^{III}(p')T_B^{III}(q') - P(\Gamma_n)r]\Big) \\
&= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\big[\big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big) \\
&\quad + \big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big) \\
&\quad - \big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)\big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big)\big] \\
&\quad - \max_{p'+q'-p'q'=r}\Big[\Big(\Big(P\big(\Gamma_{t_{12}}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_{12}}\big)P_{FP}(A_\theta)\Big) \\
&\quad + \Big(P\big(\Gamma_{t_{23}}\big)P_{TP}(B_\pi) + P\big(\Gamma_{n_{23}}\big)P_{FP}(B_\pi)\Big) \\
&\quad - \Big(P\big(\Gamma_{t_{12}}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_{12}}\big)P_{FP}(A_\theta)\Big)\Big(P\big(\Gamma_{t_{23}}\big)P_{TP}(B_\pi) \\
&\quad + P\big(\Gamma_{n_{23}}\big)P_{FP}(B_\pi)\Big)\Big)\Big]\Big) \\
&= \frac{1}{P(\Gamma_t)}\Big(\max_{p+q-pq=r}\left[h_\rho\left(T_A^W(p), T_B^W(q)\right)\right] \\
&\quad - \max_{p'+q'-p'q'=r}\left[h_\rho\left(T_A^{III}(p'), T_B^{III}(q')\right)\right]\Big)
\end{aligned}
$$

■

**Theorem 4.2.4** *(Performance Bias: Across III versus Within Label-Fused ROC curves, Boolean OR rule where p=p' and q=q') Let* $D_{A \vee B}^{III}$ *be the ROC curve for the Boolean OR label-fused across II system. Let* $D_{A \vee B}^{W}$ *be the ROC curve for the Boolean OR label-fused within system. Assume A and B are independent classification systems. Then, for a fixed false positive value max p+q-pq = r = max p'+q'-p'q', the bias between across III and within label-fused ROC curves is*

$$Bias_{D_{A \vee B}}^{III\ vs\ W} = \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[ h_\rho\left(T_A^W(p), T_B^W(q)\right) - h_\rho\left(T_A^{III}(p), T_B^{III}(q)\right) \right]$$

**Proof**: Assume that classification systems A and B are independent and the correlation coefficient, ρ, is zero. Further, assume that probabilities p and q are invariant with respect to choice of label fusion rule. Then, the bias may be expressed as:

$$Bias_{D_{A \vee B}}^{III\ vs\ W} = f_C^{Within\ OR}(r) - f_C^{Across\ III\ OR}(r)$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[ h_\rho\left(T_A^W(p), T_B^W(q)\right) - P(\Gamma_n) r$$

$$- \left( h_\rho\left(T_A^{III}(p), T_B^{III}(q)\right) - P(\Gamma_n) r \right) \Big]$$

$$= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[ \left( T_A^W(p) + T_B^W(q) - T_A^W(p) T_B^W(q) \right)$$

$$- \left( T_A^{III}(p) + T_B^{III}(q) - T_A^{III}(p) T_B^{III}(q) \right) \Big]$$

$$
\begin{aligned}
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \Big[\big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big) \\
&\quad + \big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big) \\
&\quad - \big(P(\Gamma_t)P_{TP}(A_\theta) + P(\Gamma_n)P_{FP}(A_\theta)\big)\big(P(\Gamma_t)P_{TP}(B_\pi) + P(\Gamma_n)P_{FP}(B_\pi)\big) \\
&\quad - \Big(\Big(P\big(\Gamma_{t_{12}}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_{12}}\big)P_{FP}(A_\theta)\Big) \\
&\quad + \Big(P\big(\Gamma_{t_{23}}\big)P_{TP}(B_\pi) + P\big(\Gamma_{n_{23}}\big)P_{FP}(B_\pi)\Big) \\
&\quad - \Big(P\big(\Gamma_{t_{12}}\big)P_{TP}(A_\theta) + P\big(\Gamma_{n_{12}}\big)P_{FP}(A_\theta)\Big)\Big(P\big(\Gamma_{t_{23}}\big)P_{TP}(B_\pi) \\
&\quad + P\big(\Gamma_{n_{23}}\big)P_{FP}(B_\pi)\Big)\Big)\Big] \\
&= \frac{1}{P(\Gamma_t)} \max_{p+q-pq=r} \left[h_\rho\big(T_A^W(p), T_B^W(q)\big) - h_\rho\big(T_A^{III}(p), T_B^{III}(q)\big)\right]
\end{aligned}
$$

■

The bias formulas that are concerned with the *across I* and *across III* label fusion rules are very similar in appearance. It will be seen in section V that this similarity extends to the ROC curves themselves, but the presence of the common type 2 target in *across III* fusion will lead to different results as compared to *across I* label fusion.

In the following section the simulation that was produced will be discussed and the results analyzed. Finally, in section VI, the discussion will highlight those results that were discovered through this work.

# V Simulation

A computer simulation was used to determine the extent of performance bias that may exist between classification systems that use *within* label fusion as opposed to *across* label fusion. In addition, individual classification system performance, correlation, and target prevalence were varied in order to establish how these factors affect any potential bias. All coding for the simulation was developed using *MATLAB*®.

## 5.1 Construction of the Simulation

### 5.1.1 Simulated Classification Systems

Assume that classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ exist and are the same mapping compositions as outlined in section 3.1.2. In this instance, the parameter set, π, is a two dimensional set composed of parameters γ and ε. Define the parameter sets θ = [-4, 6] and the parameters sets γ = [-4, 6] and ε = [0, 10]. Assume that the features of the non-target distribution, $\Phi_n$, is distributed as N(0, 1). Also, assume that the features of the target distribution, $\Phi_t$, is normally distributed with variance of 1. Let the two classification systems be defined thusly:

$$a_\theta = \begin{Bmatrix} t: & x \in \Gamma_t;\; x \geq \theta \\ n: & otherwise \end{Bmatrix}$$
$$b_{\gamma,\varepsilon} = \begin{Bmatrix} t: & x \in \Gamma_t;\; \gamma \leq x \leq (\gamma+\varepsilon) \\ n: & otherwise \end{Bmatrix} \qquad (5.1)$$

Hence for classifier $a_\theta$, if the element from the feature set is greater than or equal to the parameter value θ, then the classification system gives the element a “t” label. If not, the element is given an “n” label. For classifier $b_{\gamma,\varepsilon}$, if the element from the feature set is between the parameter value γ and the sum (γ + ε), then the element receives a “t” label and an “n” label otherwise.

### *5.1.2 Area under a ROC Curve*

In order to examine the effects of individual classification system performance on any bias that exists between *within* and *across* fused systems, area under the ROC curve (AUC) was used. AUC has a specific interpretation in that a perfect classification system has an AUC of 1 whereas an equivalently random classification system has an AUC = 0.5. Any ROC curve with AUC less that 0.5 performs worse than chance. Thus, by exploiting the statistical properties of the AUC, specific classification system performances may be determined. Using these properties, it will be assumed that the target and non-target partitions from the feature set come from different $N(\mu, \sigma^2)$ distributions. Assume that there are three levels of classification system performance: *good, fair, and poor.* A *good* classification system will have a corresponding AUC of 0.95, a *fair* classification system will have an AUC of 0.85, and a *poor* classification system will have an AUC of 0.75. Fixing system performance by these three levels of AUC implies specific distributions for the features. Hence, there exists a way to solve for the mean and standard deviations of the distributions for targets and non-targets that generates the ROC curve with specific AUC.

**Definition 5.1** *(Normal Area under a ROC curve) Let $a = (\mu_+ - \mu_-)$ (the difference between the mean of the target distribution and non-target distribution). Let $b = (\sigma_- / \sigma_+)$ (the ratio of the standard deviation of the non-target distribution over the standard deviation of the target distribution). Let $\Phi^{-1}$ be the inverse Normal CDF. Then the area under an ROC curve for normally distributed event partitions is*

$$AUC = \Phi^{-1}\left(\frac{a}{\sqrt{1+b^2}}\right) = \Phi^{-1}\left(\frac{(\mu_+ - \mu_-)}{\sqrt{1+\left(\frac{\sigma_-}{\sigma_+}\right)^2}}\right)$$

Let the distribution of the non-targets be N(0, 1). Thus, for a *good* classification system, $\mu_t = 2.326$. For a *fair* classification system, $\mu_t = 1.465$, and for a *poor* classification system $\mu_t = 0.954$. The classification systems defined in 5.1.1. were constructed specifically so that each of the three previously described levels of performance using AUC criteria could be established with the target means as defined above.

*5.1.3 Simulation Scenarios*

The probability of true positive (TP) and false positive (FP) for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ is defined to be the following.

$$P_{TP}(A_\theta) = P\left(A_\theta^\natural(\Lambda_t)|\Gamma_t\right) = \frac{P\left(A_\theta^\natural(\Lambda_t) \cap \Gamma_t\right)}{P(\Gamma_t)} = \int_\theta^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu_t)^2}{2}} dx$$

$$P_{FP}(A_\theta) = P\left(A_\theta^\natural(\Lambda_t)|\Gamma_n\right) = \frac{P\left(A_\theta^\natural(\Lambda_t) \cap \Gamma_n\right)}{P(\Gamma_n)} = \int_\theta^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-(x)^2}{2}} dx$$

$$P_{TP}(B_{\gamma,\varepsilon}) = P\left(B_{\gamma,\varepsilon}^\natural(\Lambda_t)|\Gamma_t\right) = \frac{P\left(B_{\gamma,\varepsilon}^\natural(\Lambda_t) \cap \Gamma_t\right)}{P(\Gamma_t)} = \int_\gamma^{(\gamma+\varepsilon)} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu_t)^2}{2}} dx$$

$$P_{FP}(B_{\gamma,\varepsilon}) = P\left(B_{\gamma,\varepsilon}^\natural(\Lambda_t)|\Gamma_n\right) = \frac{P\left(B_{\gamma,\varepsilon}^\natural(\Lambda_t) \cap \Gamma_n\right)}{P(\Gamma_n)} = \int_\gamma^{(\gamma+\varepsilon)} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x)^2}{2}} dx$$

Using *Matlab* programming software, these values and their corresponding ROC curves were generated for each of the two Boolean rules (Boolean *AND* and *OR*) and for each of the three combinations of performances for $A_\theta$ and $B_{\gamma,\varepsilon}$, (toggling between *good*, *fair*, and *poor)* as determined by AUC.  In addition to varying the performance of

competing classification systems, the effects of correlation between classification systems were examined. Seven different levels of correlation were considered; -0.8, -0.5, -0.3, 0 (independence), 0.3, 0.5, and 0.8.  It was also of interest to investigate the effects associated with altering target populations. Three target populations were investigated, a *target rich population*, a *target enhanced population*, and a *target deficient population*. In the *target rich environment*, probability of observing a target event is $P(\Gamma_t) = 4/5$ and the probability of observing a non-target event is $P(\Gamma_n) = 1/5$. In the *target enhanced environment*, the probability of observing a target event $P(\Gamma_t) = 2/3$ and the probability of observing a non-target event is $P(\Gamma_n) = 1/3$. Finally, in the *target deficient environment*, the probability of observing a non-target event is $P(\Gamma_t) = 1/5$ and the probability of observing a non-target event is  $P(\Gamma_n) = 4/5$.

### *5.1.4 Algorithm Outline*

In this section, a brief outline of the algorithm is given. Both the source code comments are given in appendix C.

1) Create parameters θ, γ, and ε

$$\theta = \mathbb{R}^{Nx1}$$
$$\gamma = \mathbb{R}^{Nx1}$$
$$\varepsilon = \mathbb{R}^{Nx1}$$

2) Generate the probabilities of true and false positive at each parameter value for individual systems $A_\theta$ and $B_{\gamma,\varepsilon}$.

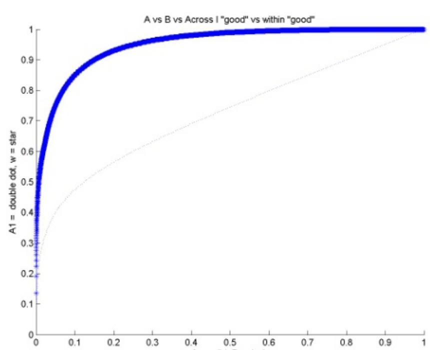
$$for\ i = 1{:}N$$
$$P_{TP}(A_\theta(i)) = \int_{\theta(i)}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu_t)^2}{2}} dx$$
$$P_{FP}(A_\theta(i)) = \int_{\theta(i)}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x)^2}{2}} dx$$
$$end$$

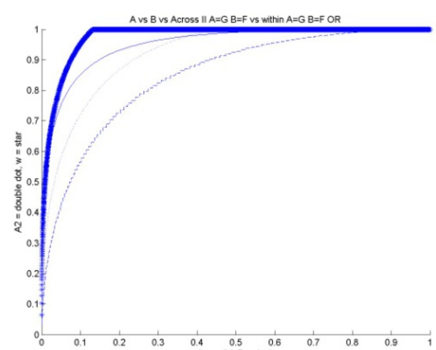
$$
\begin{aligned}
&for\ i = 1\!:\!N\\
&\quad for\ j = 1\!:\!N\\
&\qquad P_{TP}\left(B_{\gamma,\varepsilon}(i,j)\right) = \int_{\gamma(i)}^{\gamma(i)+\varepsilon(j)} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu_t)^2}{2}} dx\\
&\qquad P_{FP}\left(B_{\gamma,\varepsilon}(i,j)\right) = \int_{\gamma(i)}^{\gamma(i)+\varepsilon(j)} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x)^2}{2}} dx\\
&\quad end\\
&end
\end{aligned}
$$

3) The ROC curve for systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are computed from the probabilities of 2).

$$
\begin{aligned}
f_{A_\theta} &= \max_{\theta \in \Theta}[P_{FP}(A_\theta) \quad P_{TP}(A_\theta)]\\
f_{B_{\gamma,\varepsilon}} &= \max_{\substack{\gamma \in \Gamma' \\ \varepsilon \in E}}\left[P_{FP}\left(B_{\gamma,\varepsilon}\right) \quad P_{TP}\left(B_{\gamma,\varepsilon}\right)\right]
\end{aligned}
$$

4) Input the true positive and false positive values associated with the ROC curves for systems $A_\theta$ and $B_{\gamma,\varepsilon}$ into one of the ROC curve formulas for either *within* or *across* fusion to generate the label-fused ROC curve for the *within* and *across* combined systems.

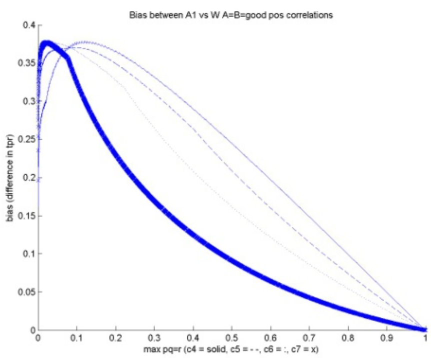


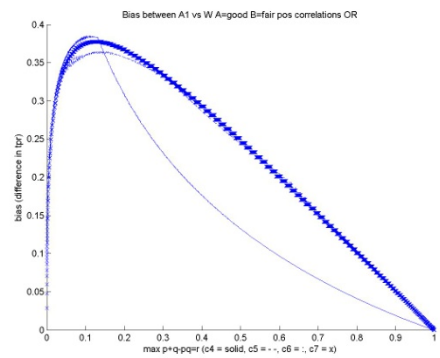




The figures above are examples of ROC curves. For the figure on the right, the disparity in true positive rate (height of the curve) is the performance bias between label-fused ROC curves. ROC curves were generated for each possible combination of performance, prevalence, and target environment. They are

supplementary material, but as the focus of this document is in the performance bias between ROC curves, the ROC curves themselves will not be discussed further here. The ROC curves are catalogued in appendix A.

5) Compute the performance bias for a given Boolean rule, a given level of performance for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$, and positive or negative correlation. Each output consists of four curves depending on whether investigating positive or negative correlation coefficients.




6) Plot *bias* versus false positive rate to create the *bias curve* between *within* and *across* label-fused ROC curves. The *bias curve* graphically depicts the difference in true positive rate at every combined false positive. The figures above are examples of these *bias curves*.

*5.2 Correlation and its Effects on Bias between Across and Within Combined Classification Systems*

In the prior section, all of the formulas for the bias were derived under the assumption that the two systems were uncorrelated. The cost functions defined in section III were derived under the assumption of a fixed level of correlation. In the case that the

two classification systems are indeed independent, the correlation coefficient, ρ, is zero. Other values of the correlation coefficient were investigated to see the effect on performance bias between label-fused ROC curves.

Recall from section III that the correlation between classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ was defined as:

$$\rho[A_\theta, B_\pi] = \frac{[C_{A\wedge B} - C_A C_B]}{\sqrt{C_A(1-C_A)}\sqrt{C_B(1-C_B)}} \qquad (5.1)$$

The correlation expression is built into the $g_\rho$ and $h_\rho$ functions (3.13 and 3.15, respectively). Six different levels of non-zero correlation coefficient values were chosen such that a trend may be seen in the corresponding bias curve output.

| |
|---|
| $\rho_1 = -0.8$ |
| $\rho_2 = -0.5$ |
| $\rho_3 = -0.3$ |
| $\rho_4 = 0.3$ |
| $\rho_5 = 0.5$ |
| $\rho_6 = 0.8$ |

Table 5-18 - Different levels of correlation to be tested.

In the context of this document, a ROC curve, regardless of the level of correlation, can never attain a true positive rate greater than the value "1." Simultaneously, a lower bound must be placed on the true positive rate when negative correlation is considered. As it is not possible for a probability to be less than zero, the

lower bound on the true positive rate is the max of either zero or the value returned by the ROC function. These bounds may be defined as:

$$\max_r\big(f_c(r)\big) = \min_r\{1, f_c(r)\}$$
$$\min_r\big(f_c(r)\big) = \max_r\{0, f_c(r)\} \qquad (5.2)$$

In the following subsections, the effects of altering probability weights for *across* specific targets, altering the level of dependence between individual classification systems, and altering target environment are investigated and their effect on performance bias is analyzed.

*5.2.1 – The Effect of Correlation on Bias between Across I and Within Combined Classification Systems*

The difference in performance between *across I* label-fused systems as compared to *within* label-fused systems can be significant. Recall that in the context of *across I* label fusion, it is assumed that there are two mutually exclusive target types that classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are tuned to classify. Under this assumption, the likelihood that both classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ label the same element in question with a target label is quite low; particularly given that those target types to which $A_\theta$ and $B_{\gamma,\varepsilon}$ are not tuned ($t_1$ and $t_2$, respectively) have been scaled to the non-target N(0, 1) distribution. On the other hand, under the assumption of *within* label fusion, the target set for both individual classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ is the same and constitutes all element with target features in the event set.

In this simulation, the probability weights associated with the *within* label-fused classification system are equivalent to the probability of observing any target event (as determined by target environment). For the *across I* label-fused classification system, it was decided that two cases should be considered: 1) the probability of observing a $t_1$ event is equal to the probability of observing a $t_2$ event 2) the probability of observing a $t_1$ event is not equal to the probability of observing a $t_2$ event. This alteration of individual target probability gives the experimenter the ability to change the performance of the *across I* system without altering the *within* label-fused system. In section 5.2, all of the cases investigated here were simulated in the *target enhanced environment*. The other two environments will be analyzed in section 5.3. Not all figures will be presented here. For a complete listing of the figures, refer to appendix B.

*5.2.1.1 – Effect on Bias when $\rho \geq 0$ under the AND rule*

Suppose that two classification systems are highly positively correlated. Given that this is the case, it would be reasonable to believe that the two classification systems under the *AND* rule may exhibit increased performance. There is a clear trend that arises from the investigation of the bias between *across I* versus *within* correlated classification systems under the *AND* rule: as the correlation between classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ increases, the performance bias decreases.

It was hypothesized that altering both target prevalence and classification system performance should have an effect on the level of bias between *within* and *across* label-fused ROC curves. Consider first the case that both classification systems have equal levels of performance (both systems are either *good*, *fair*, or *poor*) and the probability

weighting associated with *across* target types is equally distributed among the two classification systems under the *across I* label fusion rule.

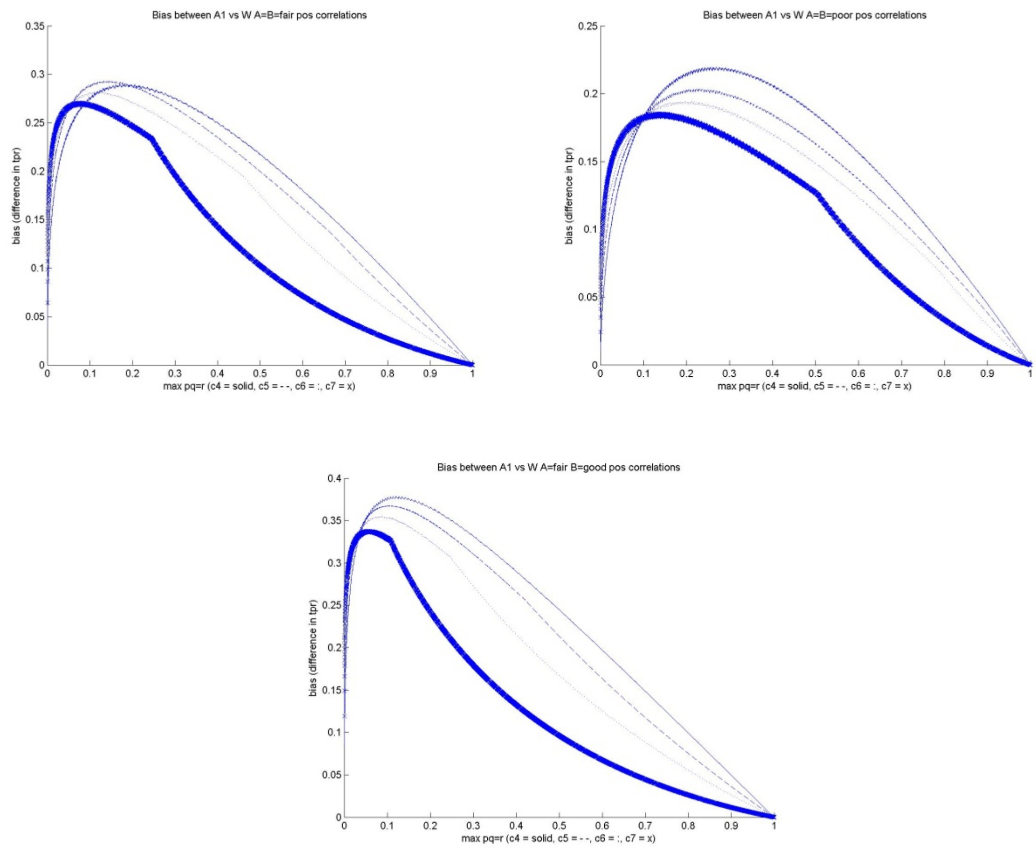



**Figure 5-1 – Bias curves *across I and* versus *within AND*. The y – axis is a measure of the bias between the two classification systems at a fixed fpr (x – axis). In the three graphs above, classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are fused at equal levels of performance. For reference, $P(\Gamma_{t1}) = P(\Gamma_{t2}) = 1/3$ and $P(\Gamma_{n1}) = P(\Gamma_{n2}) = 2/3$ under the *across I* label fusion rule.**

Altering the performance of individual classifications indeed plays a role in the magnitude of performance bias between different label-fused ROC curves. As is demonstrated in figure 5-1, as the level of performance of classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ decreases, the difference in performance between the types of label fusion also decreases. The change in performance for the *within* label-fused ROC curves is more

pronounced whereas the *across I* label-fused curves aren't performing much better than chance even at the *good* level. This makes sense for the *across I* system as $A_\theta$ and $B_{\gamma,\varepsilon}$ are tuned to mutually exclusive target types, so it is unlikely that the systems will simultaneously label any target event with the target label.

Correlation also has a clear effect on the bias between fused ROC curves. As the level of positive correlation increases (the dashed, dotted, and bold curves in the figures), notice that under the *AND* rule the level of bias between *across I* and *within* label-fused systems decreases appreciably. It is arguable that this increase in performance of both systems is the result of redundant information given that the individual classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are positively correlated.

Next, consider the case where two classification systems are being fused under the *AND* rule where the level of performance is different for each individual system.

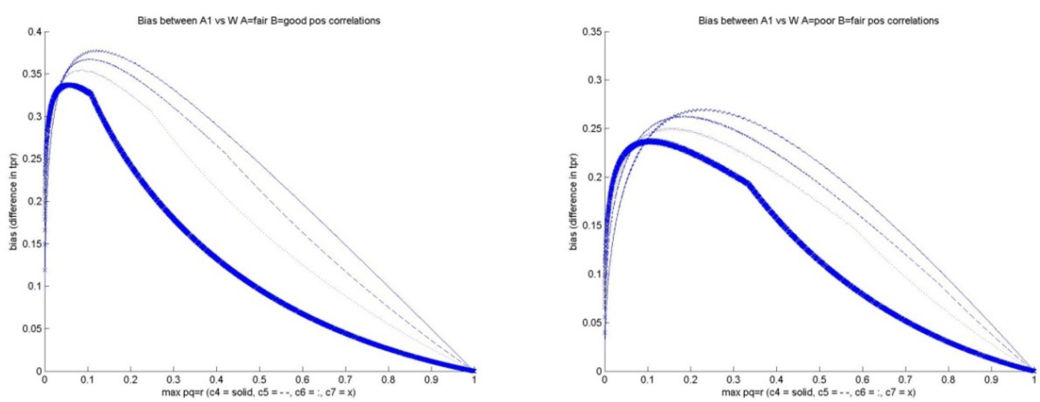



**Figure 5-2 – Fusing two classification systems that do not have equal performance. The level of bias between the fused systems is most heavily influenced by the superior individual classification system.**

Surprisingly, if difference in performance of two classification systems differs by ± 0.20 AUC, the level of bias between *across I* and *within* combined classification systems remains largely unchanged with respect to the superior classification system. For

example, consider figure 5-2 where the performance of classification system $A_\theta$ is *fair* and $B_{\gamma,\varepsilon}$ is *good*. The amount of bias between the *across I* and *within AND* label-fused ROC curves is comparable to that when both systems have *good* performance.

Next, suppose that the prior target prevalence for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ is altered such that the probability of observing a type I target is greater than the probability of observing a type II target.

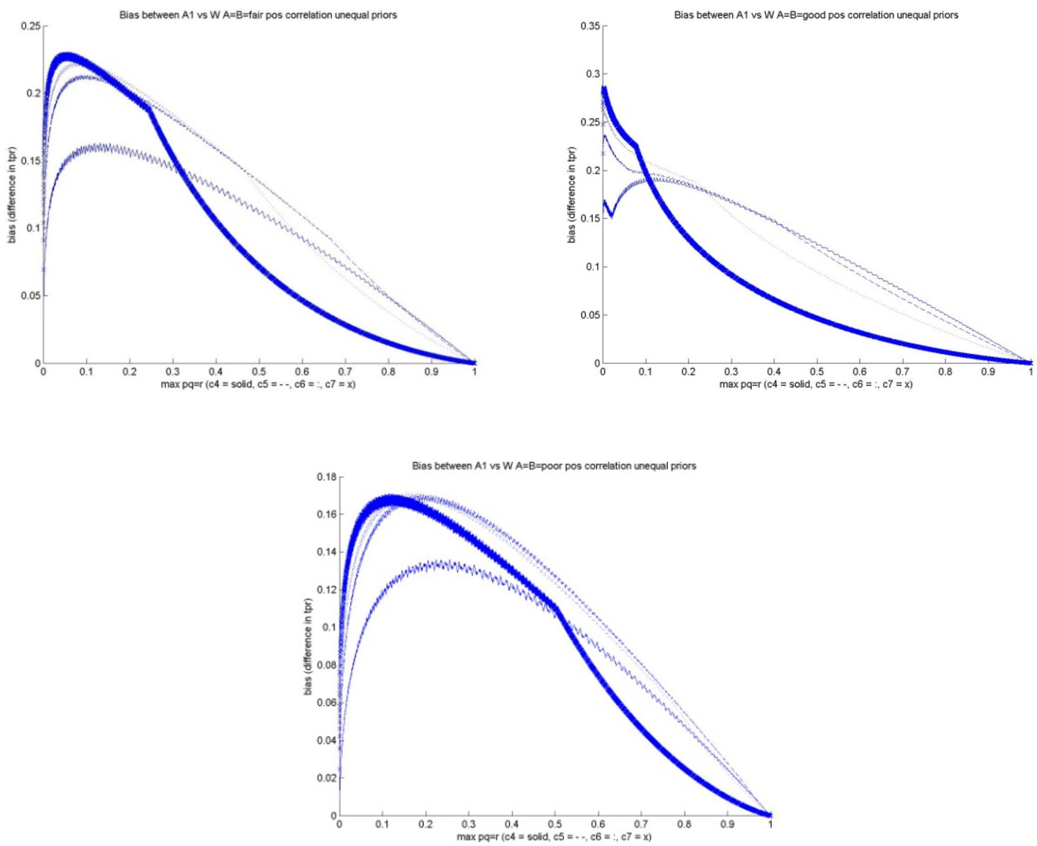



**Figure 5-3 – Altering the target prevalence for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ under *across I* label fusion has a considerable effect on the bias between systems. In this case, $P(\Gamma_{t1}) = 1/2$ and $P(\Gamma_{t2}) = 1/6$.**

Perhaps the most unusual aspect of the change in target prevalence is the effect it has on the bias curves. In the case where target prevalence was distributed equally

between the two classification systems, the shape was generally smooth and concave. The magnitude of bias between the *within* and *across I* combined classification systems under the *AND* rule decreases when performance levels of individual classification systems are equivalent. By shifting the majority of probability weight onto either $t_1$ or $t_2$ for the *across I* fused system, it is acting more like a *within* classification system. This is only true when the classification system tuned to the target type with the majority of target weight performs at a level greater than or equal to the opposing system. The opposite is true when the system with the majority of target weight performs at a lower level. In the following figure, the probability of observing a $t_1$ event is greater than the probability of observing a $t_2$ event.

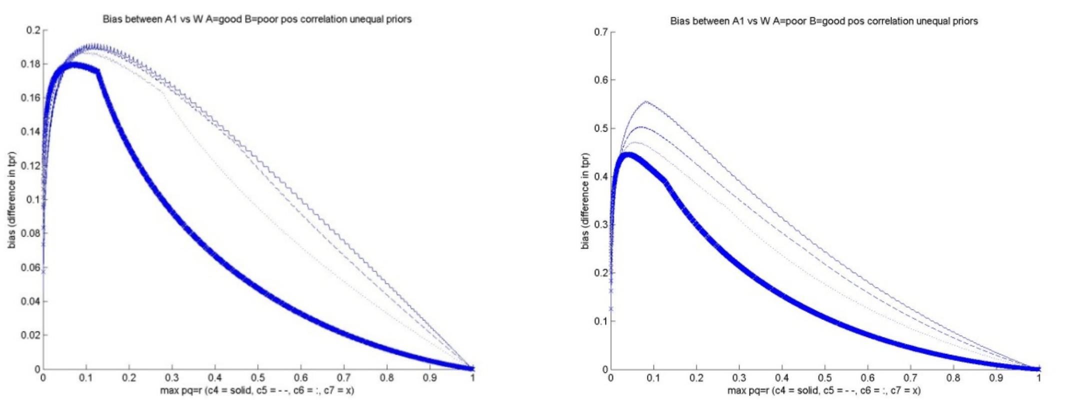



**Figure 5-4 – If system $A_\theta$ has superior performance to system $B_{\gamma,\varepsilon}$, the bias between *across I* and *within and* fused systems can be greatly decreased. If system $A_\theta$ performs poorly, regardless of the performance of system $B_{\gamma,\varepsilon}$, the fused *across I and* classification system will be severely influenced by system $A_\theta$. In this case, $P(\Gamma_{t1}) = 1/2$ and $P(\Gamma_{t2}) = 1/6$.**

If the performance of $A_\theta$ is *good* when the performance of $B_{\gamma,\varepsilon}$ is *poor* (fig 5-4, left hand side), the magnitude of bias between *within* and *across I* ROC curves reaches a maximum of roughly ±0.20 tpr ($\rho = 0$). If the performance of $A_\theta$ is *poor* when the

performance of $B_{\gamma,\varepsilon}$ is *good* (figure 5-4, right hand side) the maximum level of bias can be as high as approximately ±0.55 tpr (ρ = 0). Even in the case that both classification systems are significantly positively correlated (ρ = 0.8), the bias between the two classification systems can still be as high as ±0.40. This is to say that altering target weights for *across I* targets can be a useful tool for manipulating the performance bias.

#### *5.2.1.2 – Effect on Bias when ρ ≤ 0 under the AND rule*

When two classification systems are negatively correlated, they are said to be inversely related. First consider two classification systems with equal levels of performance and equally distributed target weight (for the *across* system). If both classification systems have *good* performance (fig 5-5, bottom), it is seen that there is only a minor relationship between negative correlation of classification systems and performance bias between *within* and *across I* label-fused ROC curves. Negative correlation has small and erratic effects on the performance bias between label-fused ROC curves under the Boolean *AND* rule. The results from the simulation suggest that the only noticeable difference in bias between *within* and *across I AND* label-fused ROC curves occurs when ρ = 0 and appears more or less constant for all tested levels of negative correlation.

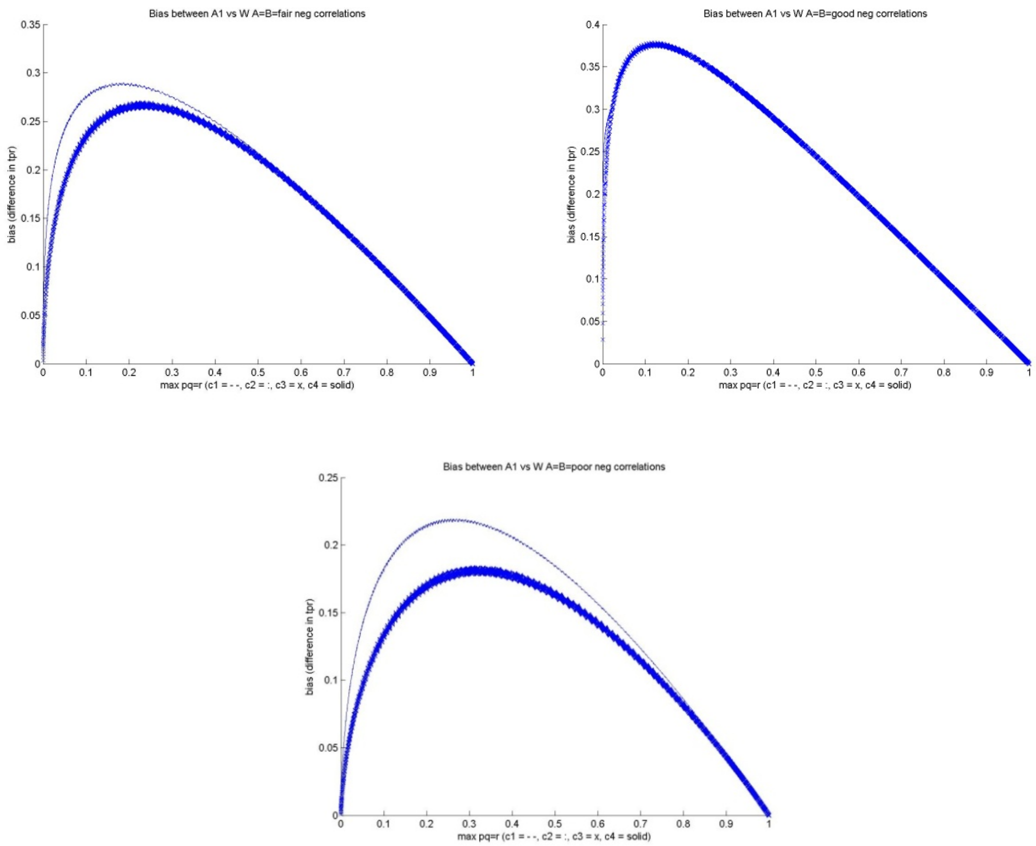



**Figure 5-5 – Negative correlation has a minor influence on the level of bias between *within* and *across I* fused classification systems. For reference, $P(\Gamma_{t1}) = P(\Gamma_{t2}) = 1/3$ and $P(\Gamma_{n1}) = P(\Gamma_{n2}) = 2/3$ under the *across I* label fusion rule.**

If the performances of individual systems are not the same, then it is reasonable to suspect that there may be a noticeable shift in the bias between classification systems. However, the simulation provided no evidence to support this hypothesis. Results from the simulation strangely suggests that when the performances of classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are different, any value from $\rho = 0$ to $\rho = -0.8$ has no effect on the performance bias between fused *within* and *across I* label-fused ROC curves (fig 5-6). It was also tested to see whether altering the weighting of *across I* target types affected bias when the individual systems were negatively correlated under the Boolean *AND* rule.

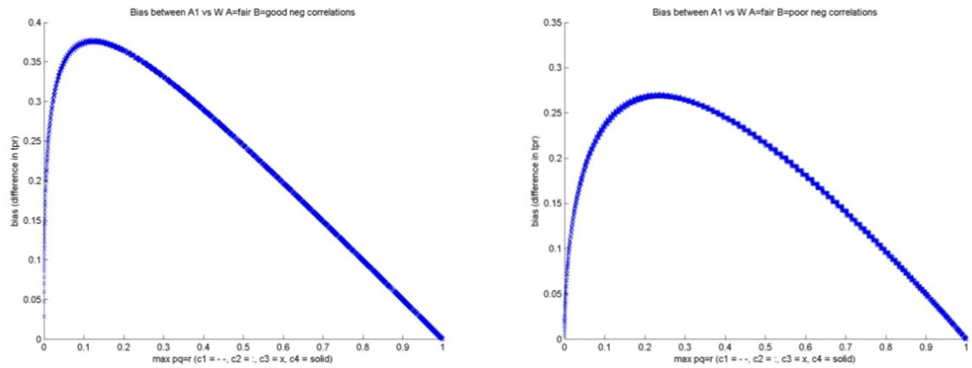
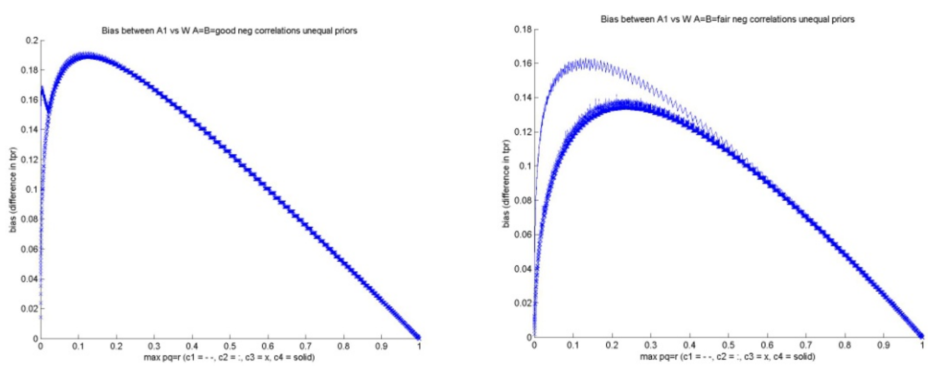



**Figure 5-6 – When two classification systems are fused with differing performance, the presence of negative correlation has little to no effect.**

Altering the weights of *across I* target types demonstrates that negative correlation does have some effect, but it is minor under the Boolean *AND* rule. In figure 5-7, it is seen that that are small differences in the performance bias when systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are performing at different levels, but hardly pronounced enough to see visibly.



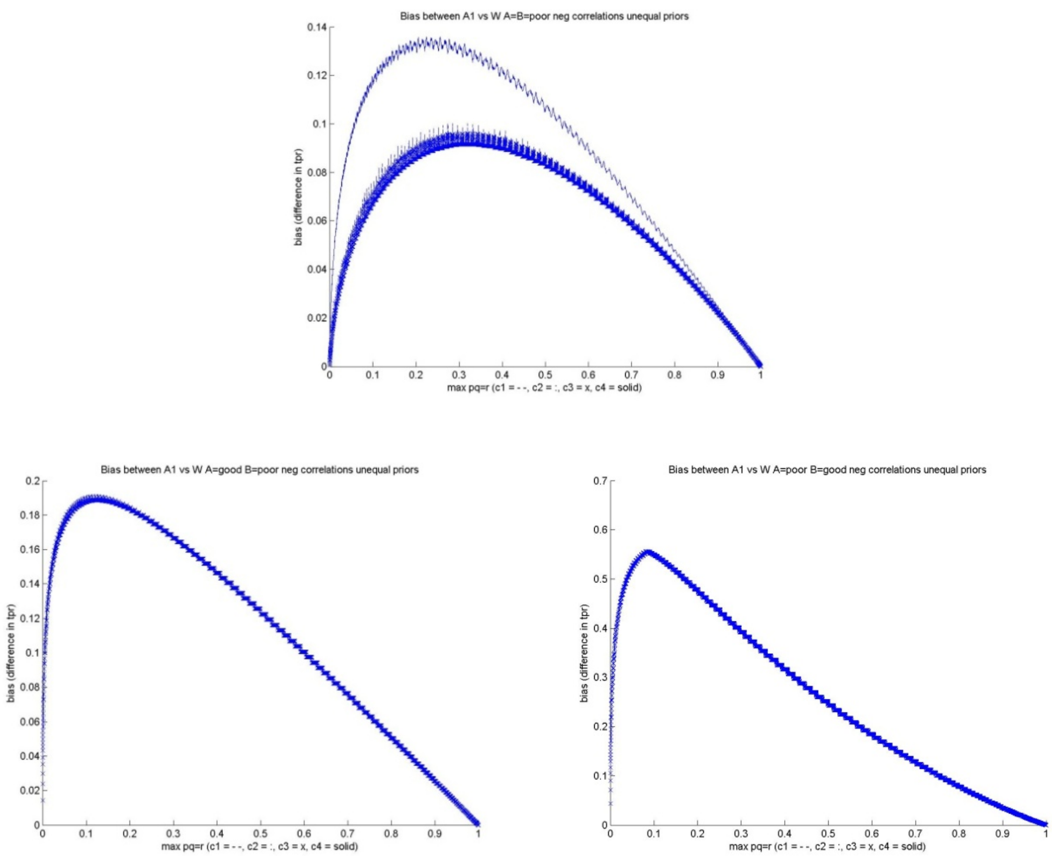



**Figure 5-7 – By increasing the prevalence of $t_1$, the influence of negative correlation becomes easier to identify. Though the effects are minimal, under the *and* rule, the level of bias does decrease between *within* and *across I* systems. In this case, $P(\Gamma_{t1}) = 1/2$ and $P(\Gamma_{t2}) = 1/6$.**

### *5.2.1.3 – Effect on Bias when $\rho \geq 0$ under the OR rule*

The results provided by the simulation suggest that positive correlation has minute effects on the bias between label-fused *OR* classification systems. This is a strange parallel with the negatively correlated label-fused *AND* curves. What can be said about the performance bias between label-fused *OR* curves is that the magnitude decreases at an accelerated rate as compared to the *AND* label-fused ROC curves. This is largely due to the fact that *within* label-fused curves under the *OR* rule reach a maximum level of performance (tpr = 1) at a lower corresponding false positive rate.

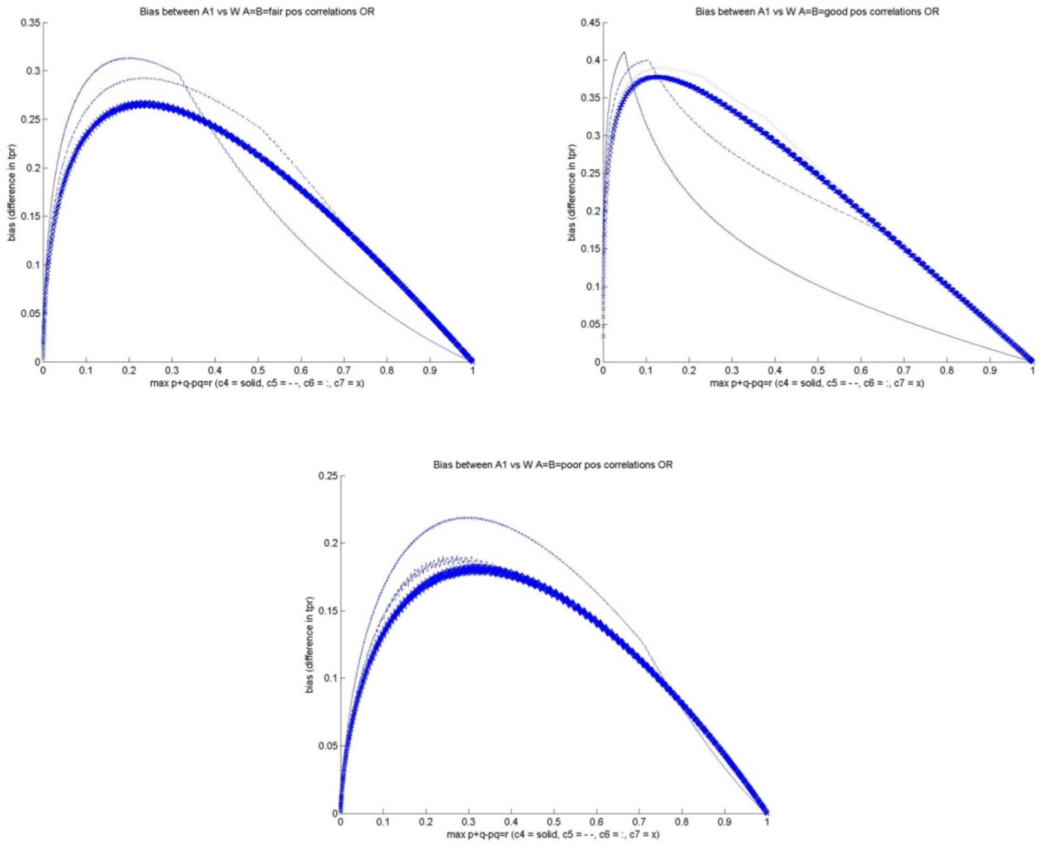



**Figure 5-8 – When the levels of performance are equal between systems $A_\theta$ and $B_{\gamma,\varepsilon}$, it appears that positive correlation negatively impacts bias. For reference, $P(\Gamma_{t1}) = P(\Gamma_{t2}) = 1/3$ and $P(\Gamma_{n1}) = P(\Gamma_{n2}) = 2/3$ under the *across I* label fusion rule.**

As seen in figure 5-8, the uncorrelated curve attains the highest magnitude of performance bias, but drops below the bias curves where systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are positively correlated under the *OR* rule. What is even more unusual is what occurs when the two classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ have different levels of performance.

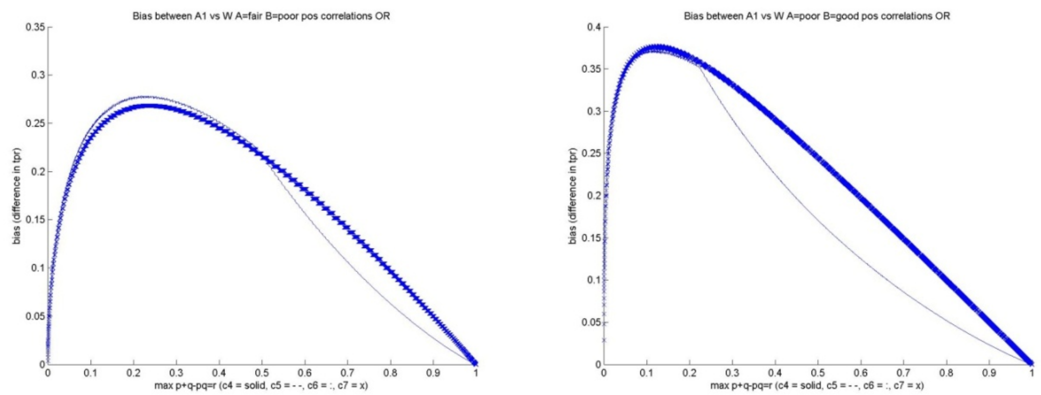



**Figure 5-9 – When systems are fused with different levels of performance, the effect of positive correlation causes an increase in bias between the *within* and *across I or* systems.**

If the performances of the two classification systems in question are different, it is interesting to note the relative lack of effect on positively correlated classification systems. From the figure above, the correlated bias curves overlap considerably while the uncorrelated curve is clearly influenced by the change in combined classification system performance. Note again that the maximum magnitude of bias seems to remain on par with the individual classification system with superior performance.

The effects of altering target weight for *across I* target types was also considered for those ROC curves fused under the *OR* rule. A trend similar to that seen under the *AND* rule can be seen in figure 5-10.

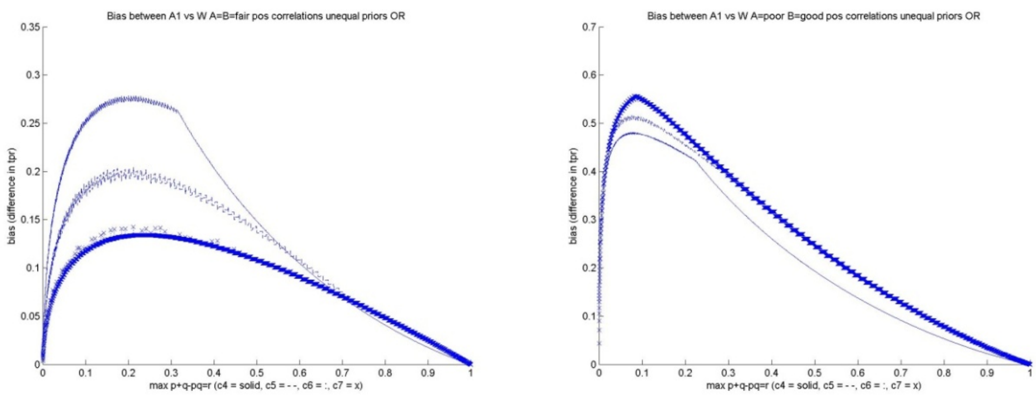



**Figure 5-10 – In the figure on the left, the uncorrelated fused classification system has the highest level of bias when both systems have *fair* performance. On the right, fusing a *poor* $A_\theta$ system with a *good* $B_{\gamma,\varepsilon}$ system in a positively correlated environment inflates the bias when positive correlation grows.**

By altering the prior probability associated with one of the targets in the *across I* combined classification system, one can increase the performance of the combined *across I OR* ROC curve if the system associated with the target that has the majority of probability weight has superior performance (fig 5-10, left). As was seen under the *AND* rule, if the system associated with the increased target weight has inferior performance, this leads to increased bias between *within* and *across I OR* ROC curves (fig 5-10, right).

*5.2.1.4 – Effect on Bias when $\rho \leq 0$ under the OR rule*

The presence of negative correlation can have a significant impact on the magnitude of bias between *within* and *across I* label-fused ROC curves under the *OR* rule. Under any level of negative correlation, the boost to performance for an *OR* ROC curve is considerably greater than when $\rho = 0$. If the performance levels are varied, this same trend continues.

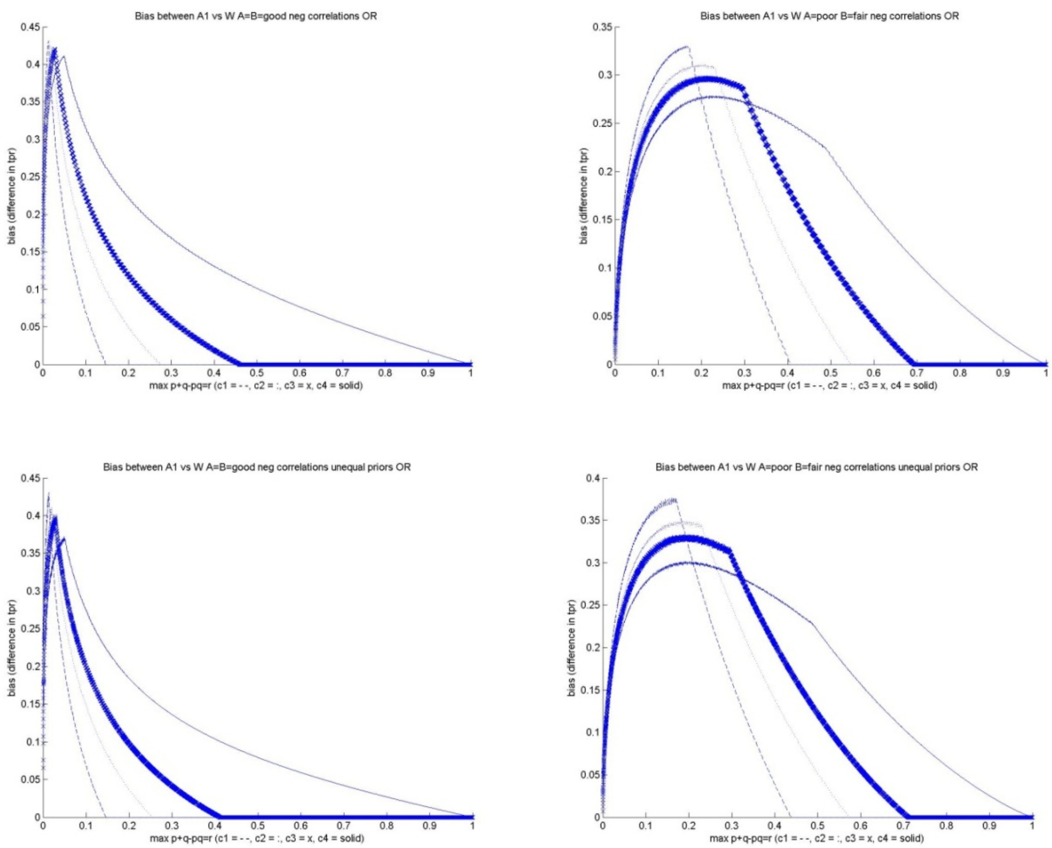



**Figure 5-11 – When the classification systems are negatively correlated (bold, dashed, and dotted curves), the bias between *within* and *across I* combined classification systems under the *OR* rule decreases. Altering the prior target prevalence of either $t_1$ or $t_2$ can exaggerate these levels of bias (bottom).**

As seen in figure 5-11, when the individual classification systems are negatively correlated and fused under the *OR* rule, the magnitude of bias between *within* and *across I* label-fused ROC curves is decreased. This trend is seen when performances of individual systems are different and when *across* target weights are altered as well.

*5.2.2 – The Effect of Correlation on Bias between Across II and Within Combined Classification Systems*

*5.2.2.1 – Effect on Bias when $\rho \geq 0$ under the AND rule*

In an *across II* environment, classification system $A_\theta$ is tuned to target type I and classification system $B_{\gamma,\varepsilon}$ is tuned to all elements in the target partition. Given that system $B_{\gamma,\varepsilon}$ classifies the same set of targets for both the *within* and *across II* systems leads to interesting results for the performance bias between *across II* versus *within* label-fused ROC curves.

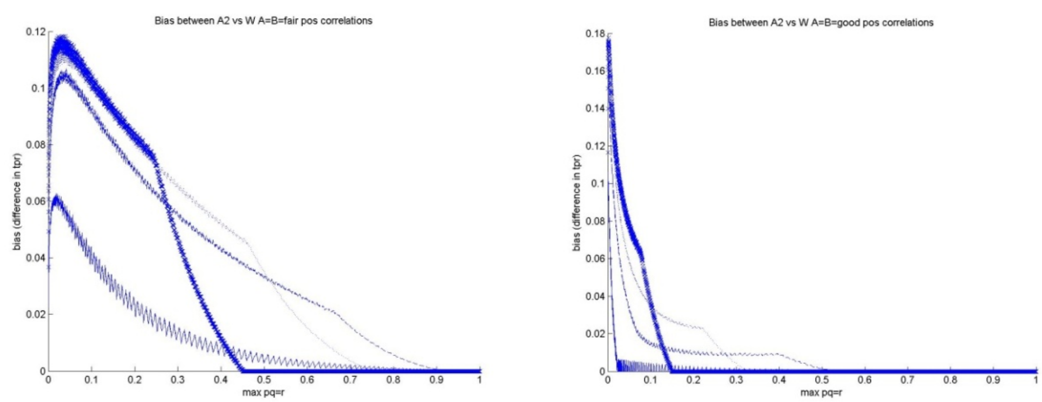
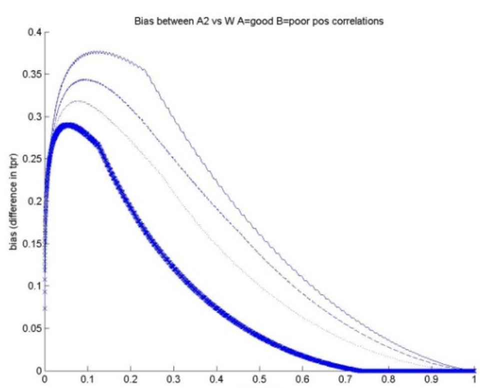
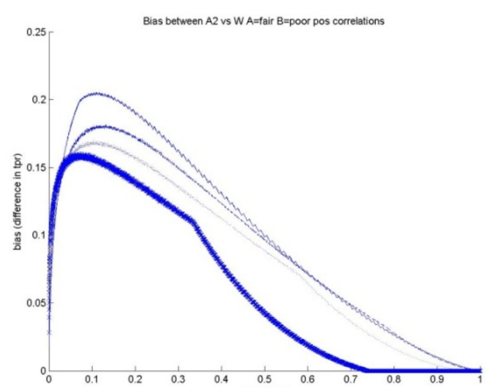


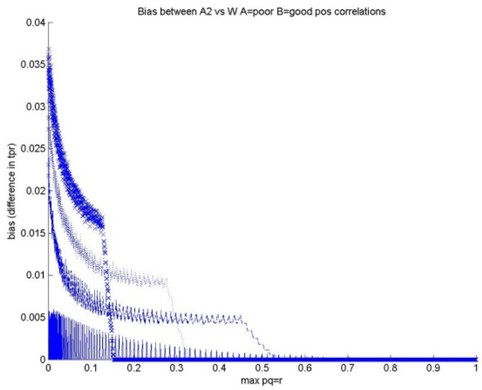

**Figure 5-12 – The magnitude of performance bias between *within* and *across II* label-fused ROC curves is considerably smaller than that seen in the previous case. This is likely caused by system $B_{\gamma,\varepsilon}$ being tuned to same partitions under both label fusion rules.**

Recall from theorem 4.2.3 that when the values of $p$ and $q$ are the same for both classification systems, the bias at a given false positive value is $\frac{1}{P(\Gamma_t)}\max_{pq=r}\left[T_B^W(q)\left(T_A^W(p) - T_A^{II}(p)\right)\right]$. Hence, the bias is largely a function of the difference in performance of classification system $A_\theta$ under the competing label fusion rules. Under the *AND* rule, the magnitude of bias between *within* and *across II* combined label-fused ROC curves is considerably smaller than that observed between *within* and

*across I* curves. This comparison highlights the relative dominance of classification systems that are designed to label all target elements in the environment. The presence of the common system $B_{\gamma,\varepsilon}$ produces cases where the level of performance for both fused systems is quite comparable. The *across II* label-fused system does not always compete on par with the *within* label-fused system. In particular, it is necessary that the performance of classification system $B_{\gamma,\varepsilon}$ is at least as good as that for system $A_\theta$. In figure 5-12, the bias curve on the right shows the output for all tested levels of correlation greater than or equal to zero. The performance bias between *within* and *across II* label-fused ROC curves under that *AND* rule drops to zero quickly for any level of positive correlation when the performance of system $A_\theta$ is equal to the performance of $B_{\gamma,\varepsilon}$ (*good* performance in this instance). However, if the performance of the two individual systems is not the same as seen in figure 5-13 (top left), the magnitude of performance bias between the two label-fused ROC curves reaches levels that were common when comparing *within* and *across I* fusion.





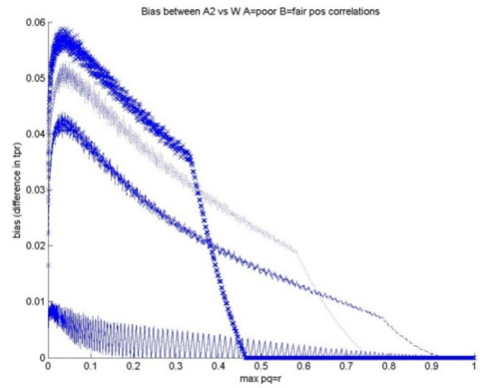




**Figure 5-13 – As demonstrated in the case of the unequal weight distribution in *across I*, if system $B_{\gamma,\varepsilon}$ performs at a lower level than system $A_\theta$, the ROC curve of the combined system is "anchored" by the performance of system $B_{\gamma,\varepsilon}$. Increasing the prior target prevalence of system $A_\theta$ (bottom graphs) further decreases the bias between *within* and *across II and* ROC curves.**

It appears that as the performance of system $B_{\gamma,\varepsilon}$ decreases, the difference in partitioning of events for classification system $A_\theta$ under the two competing label fusion rules takes precedence in determining the performance bias between the two systems.

It is important to note that under the *AND* rule, positive correlation between individual classification systems appears to increase the magnitude of performance bias between label-fused ROC curves when system $B_{\gamma,\varepsilon}$ is operating at an equal or higher level of performance than system $A_\theta$. This is unexpected given the results that are seen in the *across I* and also the *across III* cases where under the *AND* rule, positive correlation decreases the magnitude of performance bias between *within* and *across* systems regardless of performance levels. Though difficult to see, in figures 5-12 and 5-13, if system $B_{\gamma,\varepsilon}$ is operating at an equal or superior level of performance, the uncorrelated ROC curves show the lowest amount of performance bias. However, this does make sense. Though the performance of system $B_{\gamma,\varepsilon}$ is scaled equivalently for both *within* and

*across II* label fusion, the difference in performance of system $A_\theta$ becomes more drastic between the two label fusion rules, leading to increased performance bias between the two systems.

### *5.2.2.2 – Effect on Bias when $\rho \leq 0$ under the AND rule*

Negative correlation appears to have small and unpredictable effects on the magnitude of performance bias between label-fused *within* and *across II* ROC curves. This parallels quite nicely with the effects that were documented when investigating the same scenario for *across I* and *within* label-fused ROC curves under the *AND* rule.

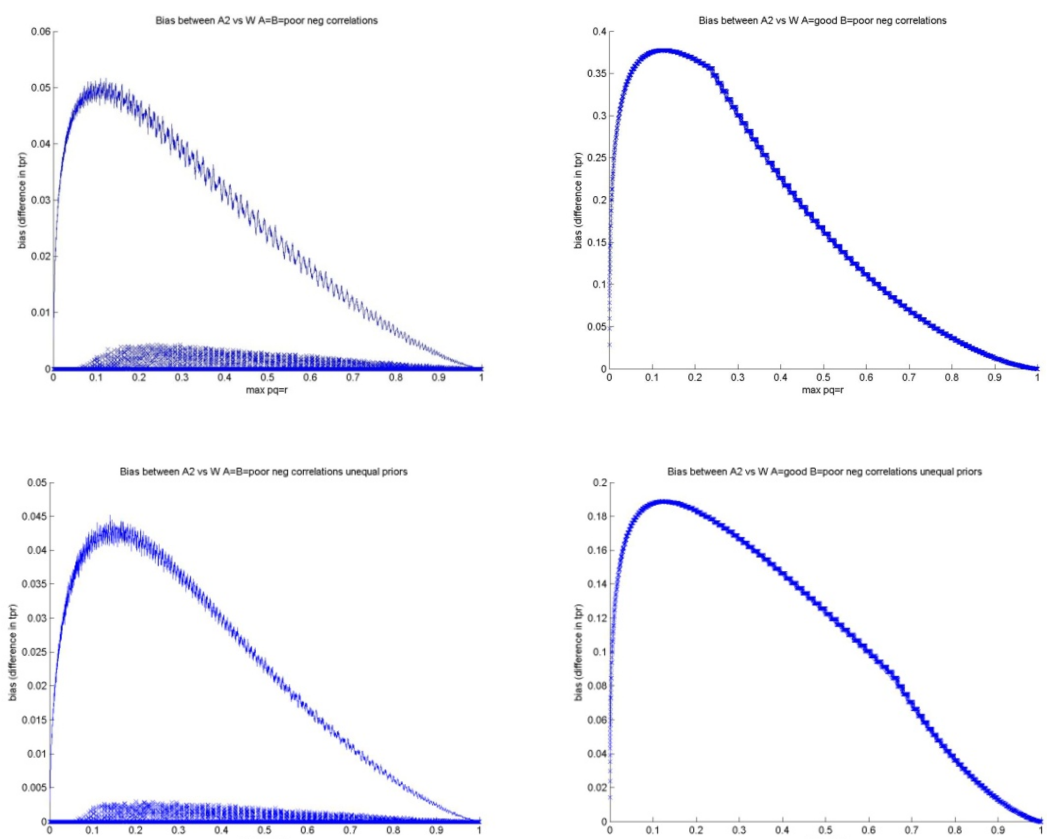



**Figure 5-14 – Negative correlation has little to no effect on the bias between *within* and *across II* label-fused systems under the *AND* rule. For bottom pictures, $P(\Gamma_{t1})$ =1/2 and $P(\Gamma_t)$ = 2/3.**

Figure 5-14 highlights four examples of the effects (or lack thereof) on bias when altering certain properties of the environment. On the left hand side, when both individual classification systems are operating at a poor performance level, the uncorrelated curve is well above the negatively correlated curves. Notice that on the bottom, increasing the likelihood of a $t_1$ event reduces the magnitude of bias in both situations. It is more clearly pronounced when altering the performance of systems $A_\theta$ and $B_{\gamma,\varepsilon}$.

#### *5.2.2.3 – Effect on Bias when $\rho \geq 0$ under the OR rule*

When comparing label-fused ROC curves for the *within* and *across I* systems under the *OR* rule, it was noted that positive correlation had small and unusual effects and negative correlation produces a clear trend of decreasing performance bias. This same general trend is seen again here.

In figure 5-15, the focus is on those instances where the levels of performance of individual systems $A_\theta$ and $B_{\gamma,\varepsilon}$ is different. It is seen that when system $B_{\gamma,\varepsilon}$ has considerably superior performance, the effect of positive correlation is hardly noticeable as the maximum magnitude never exceeds 0.045 tpr. On the right hand side, switching the performance of the systems inflates the bias. Hence, it is seen again that as long as system $B_{\gamma,\varepsilon}$ is performing at an equal or superior level of performance, positive correlation decreases bias and the opposite is true when the roles are reversed.

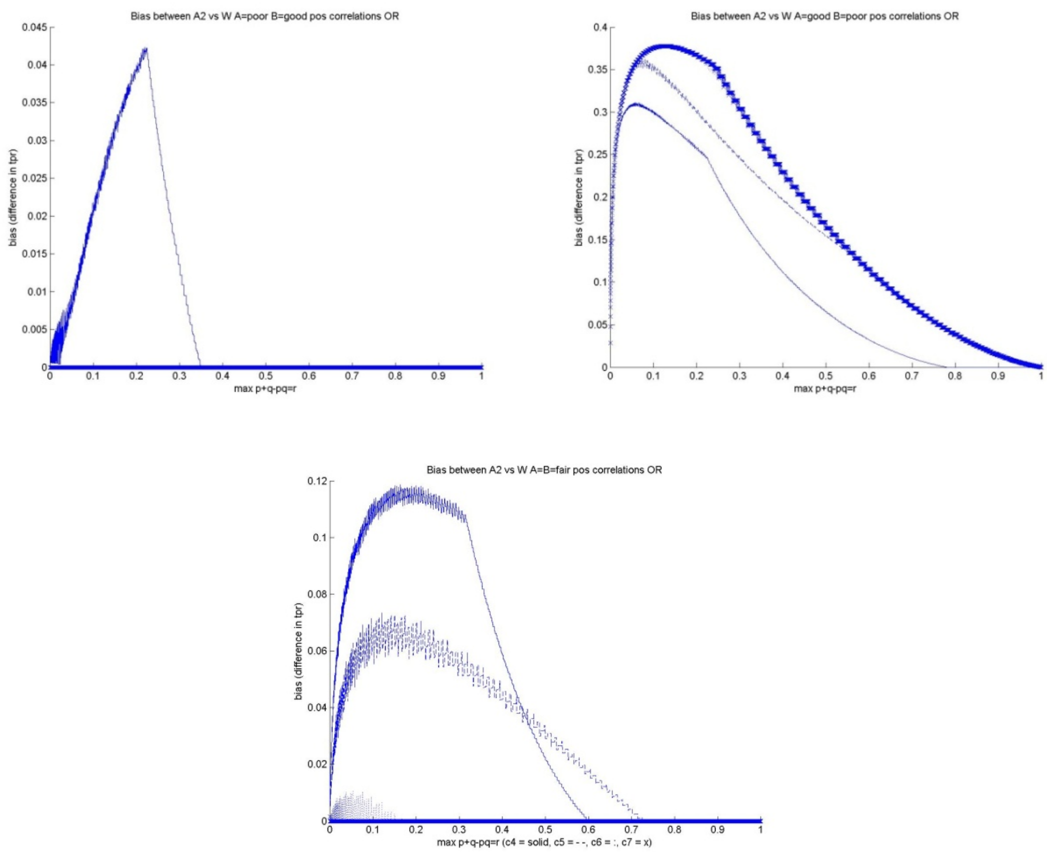



**Figure 5-15 – Notice the varying magnitudes of bias when classification system $B_{\gamma,\varepsilon}$ is respectively *good* and *poor*. If $B_{\gamma,\varepsilon}$ has good performance, the performance of system $A_{\theta}$ hardly matters (max bias of approximately 0.04). Conversely, if the performance of $B_{\gamma,\varepsilon}$ is *poor*, the bias between *within* and *across II* classifications systems can be considerable. Finally, on the bottom is the bias curve when both systems perform at the *fair* level.**

#### *5.2.2.4 Effect on Bias when $\rho \leq 0$ under the OR rule*

Given the results that were seen in section 5.2.1.4, it is anticipated that in the presence of negative correlation, the bias between *within* and *across II* combined systems will decrease when the correlation coefficient is negative. Indeed, this is the case.

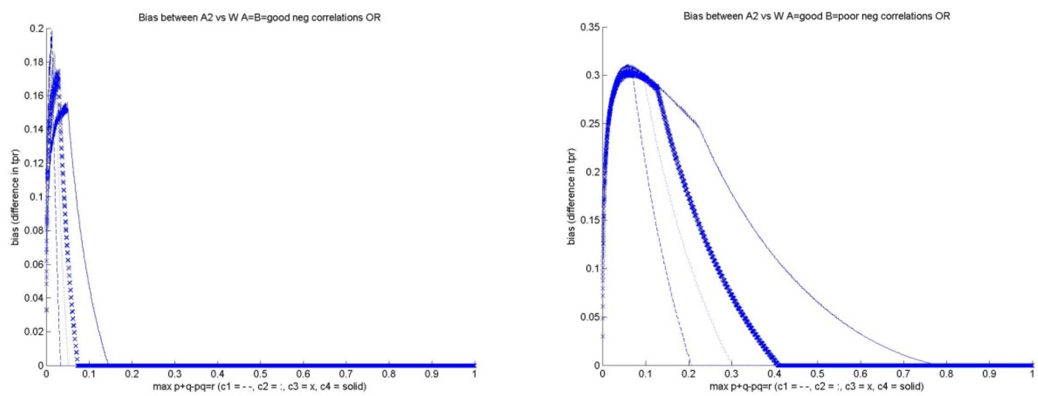



**Figure 5-16 – When ρ ≤ 0 the magnitude of bias between *within* and *across II* label-fused ROC curves under the *OR* rule decreases.**

It was demonstrated in (*Won, Cho: 2003*) that correlation can impact the performance of classification systems. Although that particular document was not concerned with label fusion, it is seen here that the boost to performance of the fused systems is causing the two ROC curves to become closer in value. Some of this bias may appear artificial as this perceived boost is being dramatized by setting a maximum true positive rate of 1, however, if the two curves are permitted to break the measure of the set, it can be seen that the relative difference in performance is indeed decreasing. The sharp decline in bias seen in the bias curves under the *OR* rule is caused by the *within* label-fused system reaching its max true positive rate and remaining constant as the *across II* label-fused ROC curve converges to the same maximum true positive rate.

As seen in figures 5-16 and 5-17, the level of performance of system $B_{\gamma,\varepsilon}$ has the most dramatic effect on magnitude of bias between the two label-fused ROC curves. This magnitude is cut nearly in half when increasing the probability of observing a $t_1$ event for system $A_\theta$.

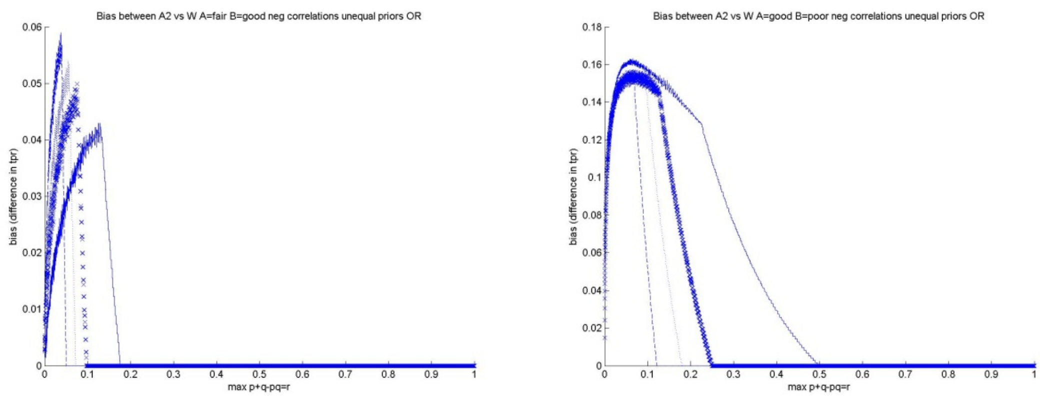



**Figure 5-17 – Increasing the prevalence of $t_1$ in an *across II* label-fused system decreases the bias. In this instance, $P(\Gamma_{t1}) = 1/2$ and $P(\Gamma_t) = 2/3$.**

### *5.2.3 – The Effect of Correlation on Bias between Across III and Within Combined Classification Systems*

The effect of correlation on the bias between *across III* label-fused systems and *within* label-fused classification systems shares many similarities with the bias between *within* and *across I* systems. This largely has to do with the fact that the target sets for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ have one mutually exclusive target type ($t_1$ and $t_3$ for classification system $A_\theta$ and $B_{\gamma,\varepsilon}$, respectively) and both also label the same subset of targets, $t_2$. This decreases the bias between label-fused ROC curves as compared to *across I* where the target sets for both classification systems were mutually exclusive.

#### *5.2.3.1 – Effect on Bias when $\rho \geq 0$ under the AND rule*

Consider the event where the prior probability of observing a target event is equal for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ under the *across III* label fusion rule. In the case that two classification systems are positively correlated under both the *within* and *across III* label rule, it is seen that the magnitude of bias decreases. This is to be expected given the similarity to the *across I* scenario

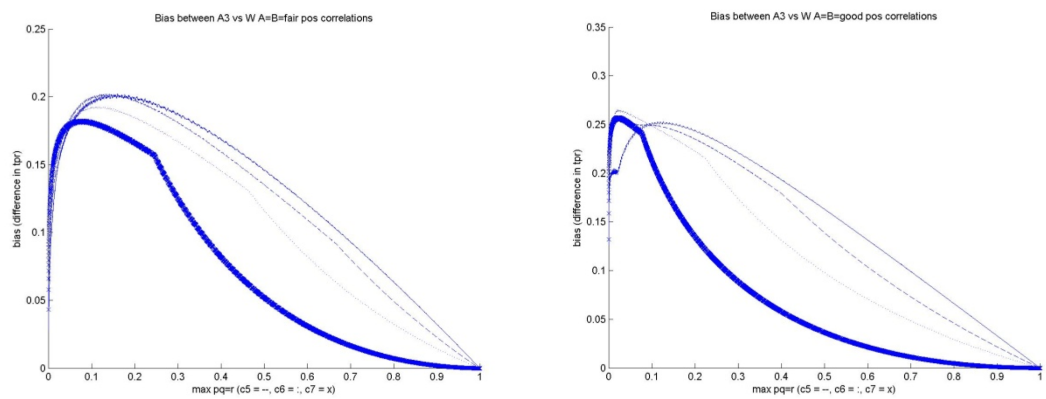



**Figure 5-18 – When ρ > 0 (bold, dashed, and dotted curves), the bias between combined *within* and *across III* systems under the *AND* rule is decreased.**

When the correlation coefficient takes on values 0.3 and 0.5 (dashed and dotted curves respectively in figure 5-18), the reduction in performance bias is minimal compared to the drastic case when the coefficient takes the value 0.8 (bold x curve). The change in performance bias is muted in some respects in this scenario likely because of the presence of the common $t_2$ subset for both systems. As noted earlier, the magnitude of bias appears to be most closely related to the classification system with the superior level of performance. This is a trend that has persisted through all tested scenarios and appears again in figure 5-19 where the levels of performance of individual systems was altered.

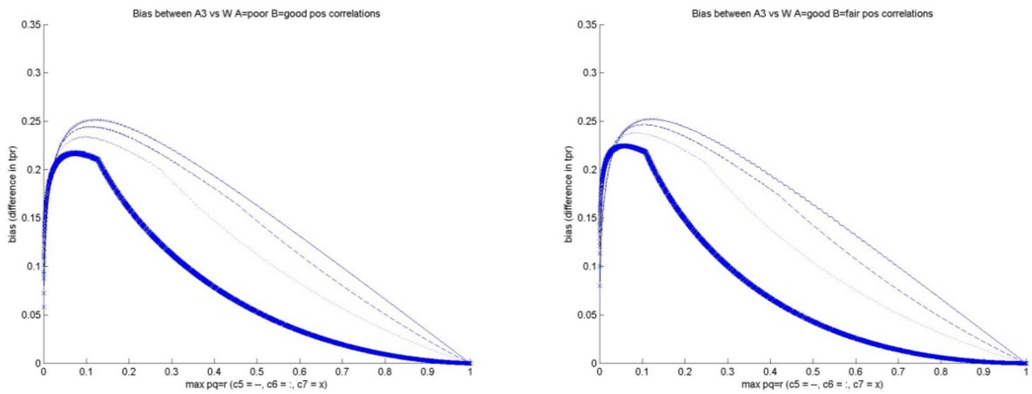



**Figure 5-19 – Combining two individual systems of different performance under the *across III* and *within* label fusion rules.**

Consider altering the target weight of observing a $t_{12}$ event to $P(t_{12}) = \frac{17}{20}$ and the probability of observing a $t_{23}$ event to $P(t_{23}) = \frac{3}{20}$ for the *across III* combined classification system under the *AND* rule.

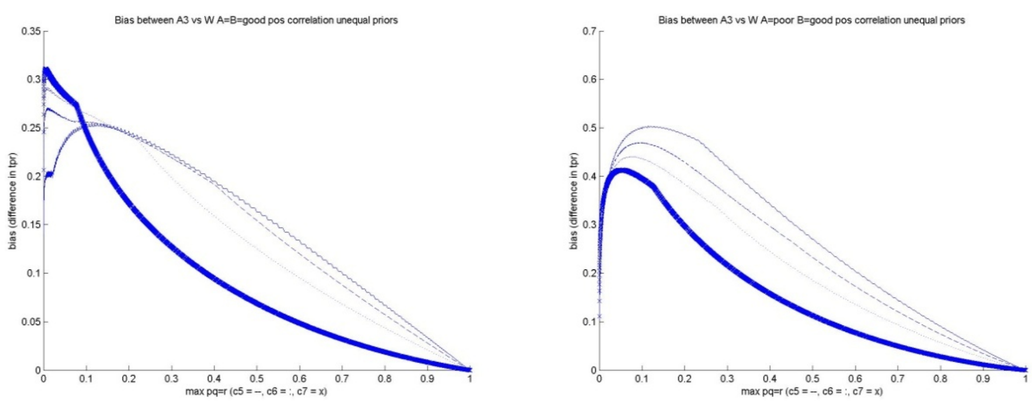



**Figure 5-20 – The bias decreases when the individual classification system with the majority of prior target prevalence has a higher or equivalent level of performance. The opposite is true when the system with the majority of target prevalence has poorer performance.**

Not surprisingly, the bias curves output when quantifying the bias between *across III* and *within* ROC curves strongly resemble the results of section 5.2.1

*5.2.3.2 – Effect on Bias when $\rho \leq 0$ under the AND rule*

When the two individual classification systems are negatively correlated under the *AND* rule, the bias curves overlap considerably. It is unclear why negative correlation seems to affect both *within* and *across III* fused systems equally (that is, there is no difference in performance for the *within* and *across III* label-fused ROC curves).

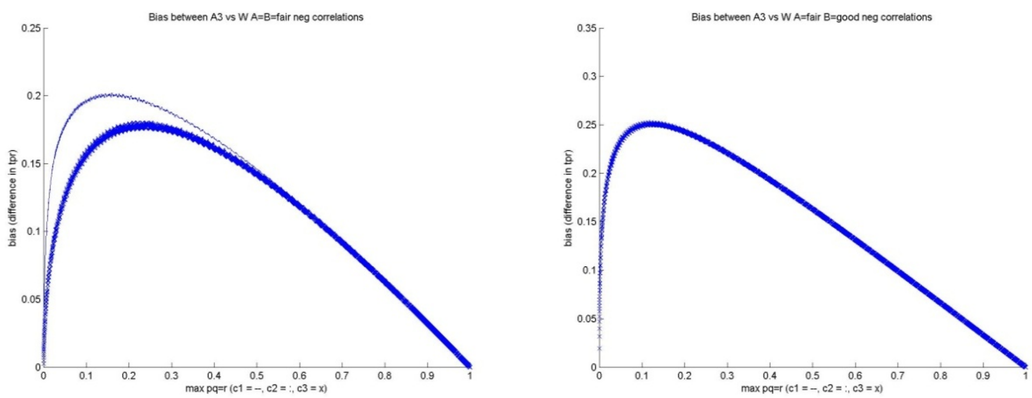



Figure 5-21 – Negative correlation has little effect on combined *AND* classification systems.

Altering prior target prevalence can be beneficial to decreasing the difference in responses between *within* and *across III* systems when the system responsible for classifying the majority of elements performs at a level equal to or superior to the other system. The converse is true when the system tasked with classifying the majority of elements in the environment has inferior performance.

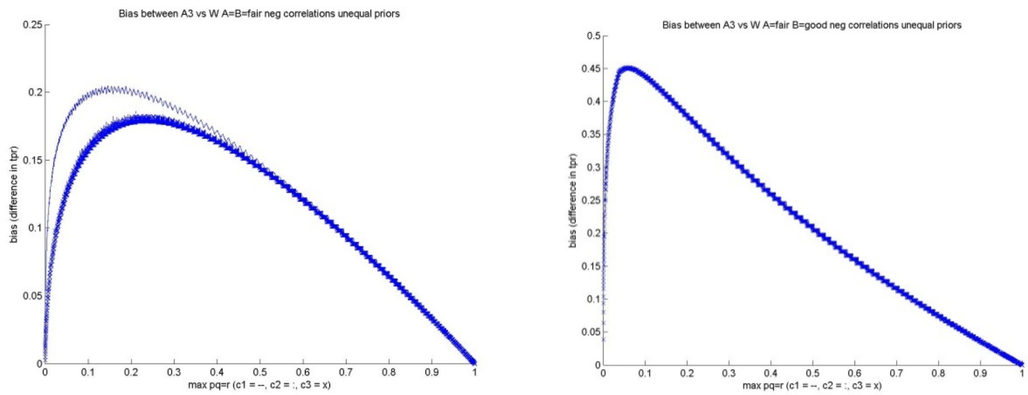



**Figure 5-22 – Oddly, combining systems of different performance seems to eliminate fluctuations in bias under the *AND* rule.**

### *5.2.3.3 – Effect on Bias when $\rho \geq 0$ under the OR rule*

Under the *OR* rule, if the two individual classification systems are positively correlated, then the closer the two systems are to being independent, the less performance bias exists. This makes sense as this same trend occurred when investigating the performance bias between *across I* and *within* label-fused ROC curves under the *OR* rule when the correlation coefficient was non-negative.

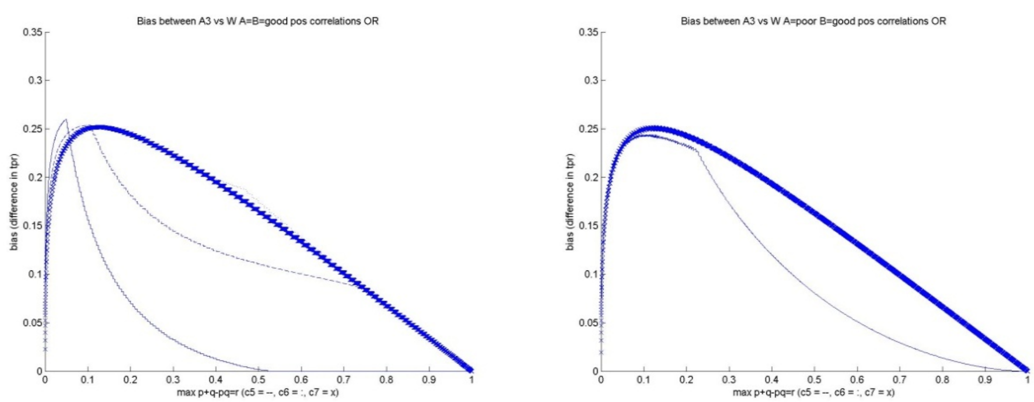



**Figure 5-23 – When the two classification systems are independent, the bias between *across III* and *within* label-fused ROC curves is minimized under the *OR* rule and the correlation coefficient is non-negative.**

Not depicted here is the altering of individual target prevalence for *across III* target types. The trend is the same as that seen in the previous two subsections and will not be discussed here to avoid redundancy.

### *5.2.3.4 – Effect on Bias when ρ ≤ 0 under the OR rule*

When the correlation constant is negative, the results are what one would expect: negative correlation under the *OR* rule implies decreased bias between *within* and *across III* combined classification systems. The magnitude of performance bias between *within* and *across III* systems is less than that observed in the *across I* scenario due to the common $t_2$ subset.

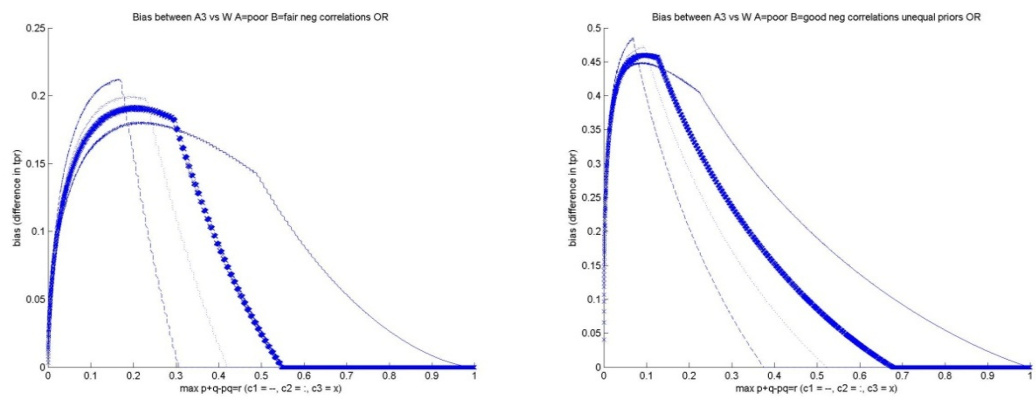



**Figure 5-24 – Negative correlation leads to "improvement" in the performance of the combined *OR* ROC curves.**

## *5.3 – Altered Target Environments and its Effects on Bias*

Recall that there are three different types of target environments being investigated in this simulation. The first target environment, the *target enhanced environment*, has been expounded upon in the section above. In a target enhanced environment, the proportion of targets to non-targets is 2/3 to 1/3. The other two target environments, the *target rich* and *target deficient* environments will be summarized here. Recall that in a *target rich environment*, the proportion of targets is 4/5 to 1/5 and in a

*target deficient environment*, the proportion of targets to non-targets is 1/5 to 4/5. Note that only the overall probability of observing a target or non-target event is being changed here.

*5.3.1 – Altered Target Environment Across I versus Within*

Depending on the type of target environment, the resulting difference in bias can be quite drastic. In a target rich environment, the bias between *within* and *across I* classification systems under the *AND* rule is increased marginally. The type of environment does not appear to change the trends that were seen in the *target enhanced environment*, though the magnitude of bias may be affected. This is most notably seen in the *target deficient environment*. It appears that reducing probability of observing a target event in an environment has the greatest effect on bias between *within* and *across I* combined classification systems (figure 5-25 middle row). In the *target rich environment*, the shape of the bias curve resembles that of the *target enhanced environment*, but notice that the curve appears stretched over the x-axis. This occurs as the false positive rate at which these levels of performance are being registered is increasing. That is to say, the higher the probability of observing a target event, the more likely it becomes that the individual classification systems incorrectly label targets with the non-target label. Therefore, it makes logical sense in an environment with proportionally few target events that classification systems with *good* performance would correctly label the true target events with the target label very effectively. That is to say, given that it is unlikely to observe a target event, the probability that a target is mislabeled is proportionally smaller. Even at a modest positive correlation of 0.3 (dashed line), the bias between *across I* and

*within* classification systems drops off tremendously under the *AND* rule (figure 5-25). Increasing this constant only drives the bias to zero at a faster rate.

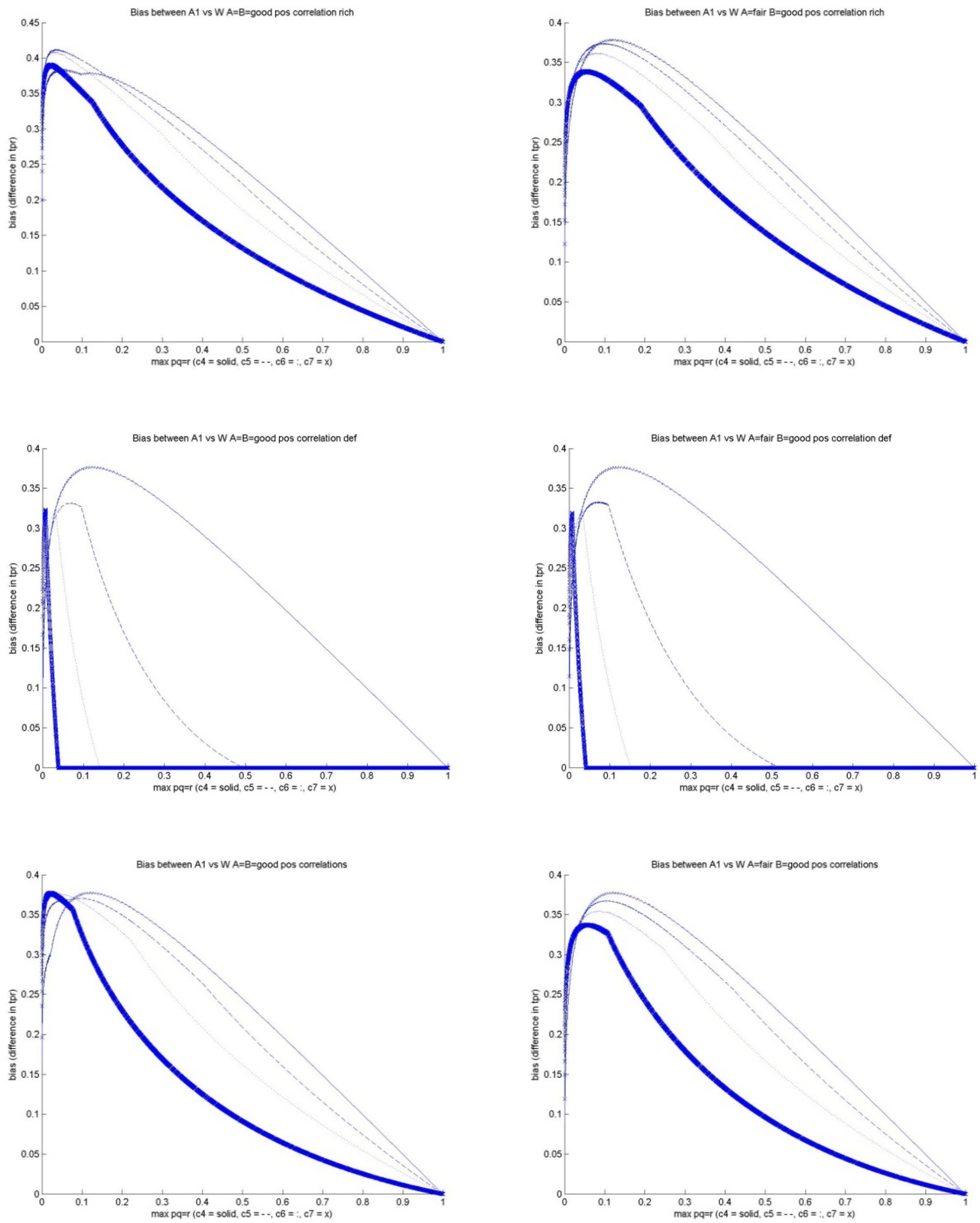



**Figure 5-25 – Altering the distribution of targets and non-targets across the entire event set has huge implications for the performance of combined systems. In a *target deficient environment*, the bias between *within* and *across***

**systems approaches zero in the presence of positive correlation under the *AND* rule. In a *target rich environment*, the overall bias increases between *within* and *across* classification systems.**

Altering the prior target prevalence for *across* type targets in different environments for systems combined under the *AND* rule also produces similar results as were seen in section 5.2 (figure 5-26). Negatively correlated classification systems under the *AND* rule and positively correlated classification systems under the *OR* rule will not be discussed here. If the reader wishes to visually inspect these cases, he or she is directed to the contents of appendix B.

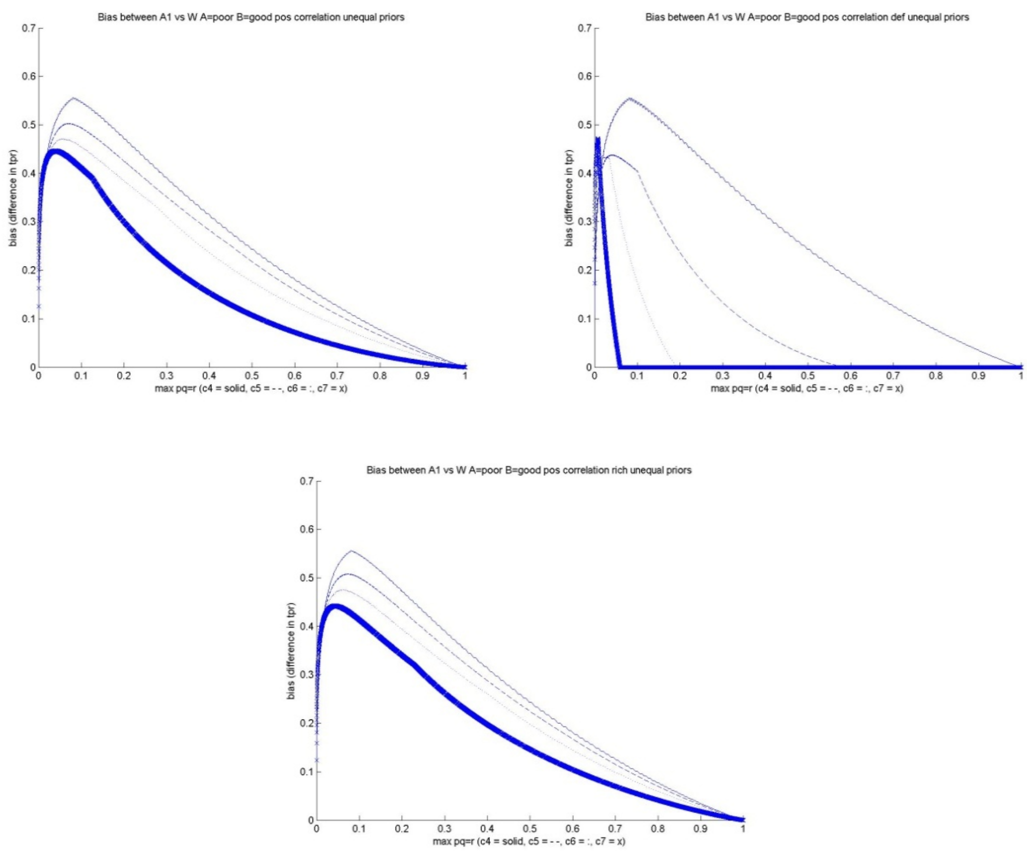
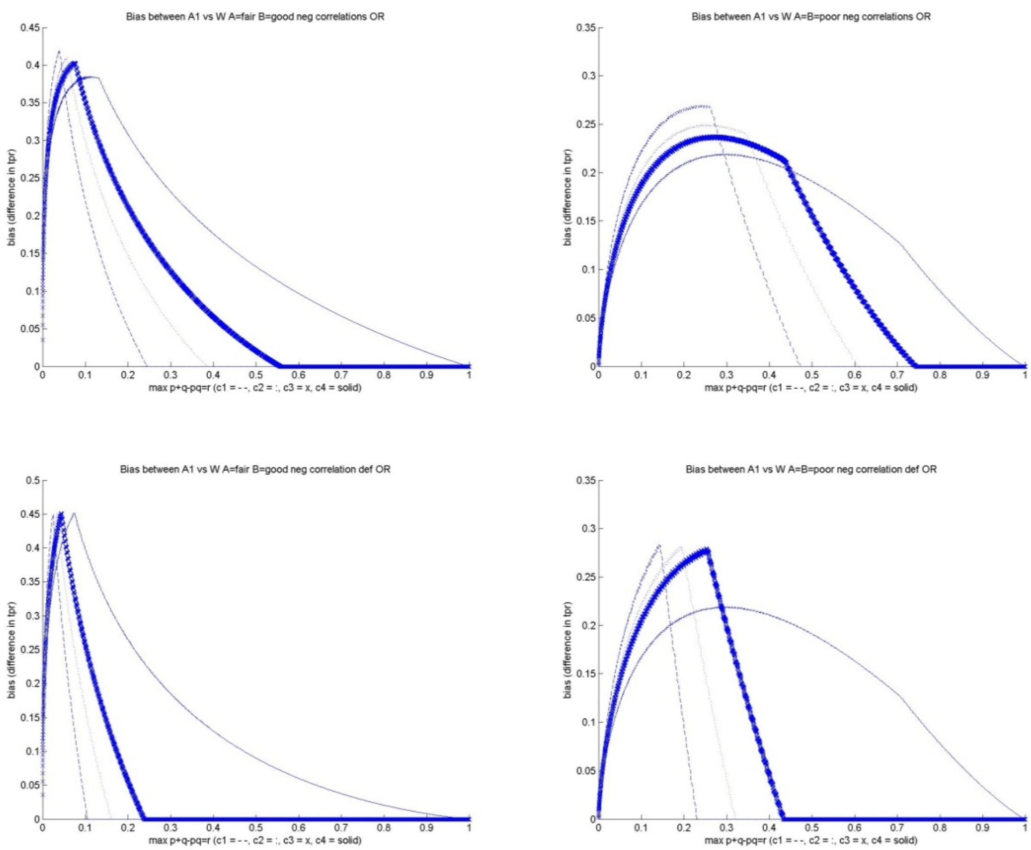



**Figure 5-26 – Altering individual target prevalence produces similar results regardless of target environment.**

Consider the label-fused ROC curves under the *OR* rule in these altered environments. As suggested before, by reducing the probability of a target event occurring, the relative performance of *within* and *across* label-fused classification systems at correctly labeling their respective target types should increase. Furthermore, if the two classification systems are negatively correlated under the *OR* rule, this boost to performance can be dramatic (figure 5-27, middle row).



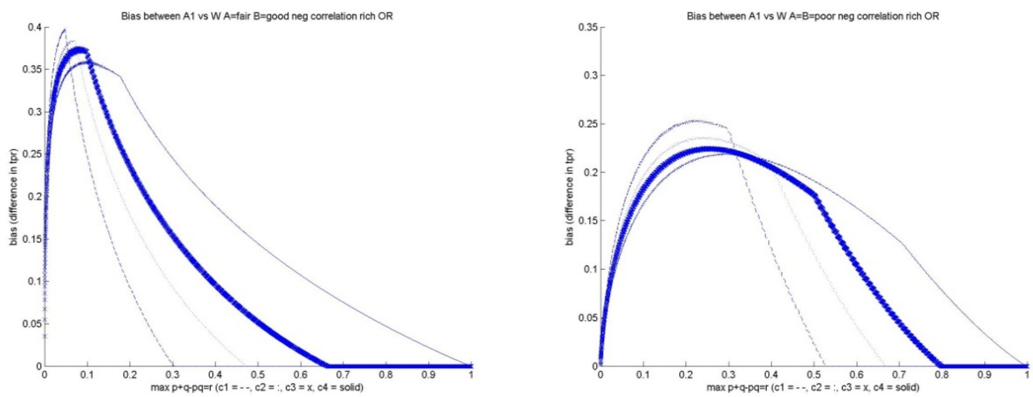



**Figure 5-27 – As was observed under the *AND* rule, the relationship between performance bias and probability of observing a target event in the environment in inversely proportional. That is to say, by increasing the probability of observing a target event, the individual classification systems are more likely to incorrectly label targets with a non-target label.**

As there are no clear trends that can be analyzed when two classification systems are positively correlated under the *OR* rule, these results will not be discussed here. The bias curves are included in appendix B if the reader wishes to observe the output for him or herself.

*5.3.2 – Altered Target Environments Across II versus Within*

Altering the target environment when comparing *across II* label-fused ROC curves to *within* label-fused ROC curves seems to dramatize the results from section 5.2. Decreasing the number of targets in the event set causes the bias between the systems to decrease more sharply and increasing the number of targets in the environment increases the bias between systems. Altering the ratio of $t_1$'s for classification system $A_\theta$ has the same effects as before, but is not pictured here. These figures have been included in appendix B.

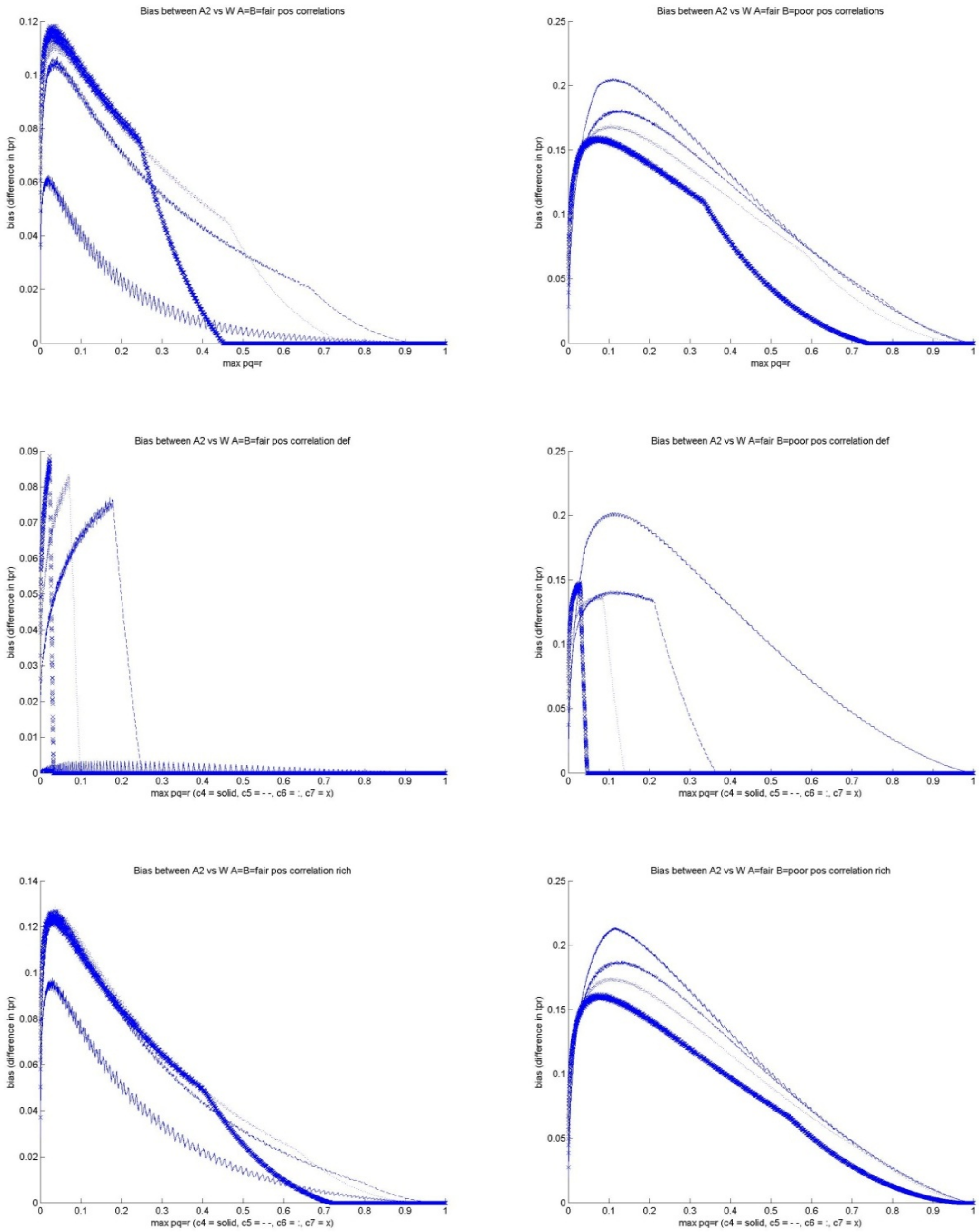



**Figure 5-28 – Decreasing the proportion of targets in the environment greatly affects bias between *within* and *across II* label-fused systems under the *AND* rule.**

As seen above, the proportion of target in the environment is inversely proportional to the magnitude of bias that occurs between *within* and *across II* label-fused

ROC curves under the *OR* rule when the correlation coefficient is negative. Positive correlation under the *OR* rule is not discussed here, but this can be investigated in appendix B.

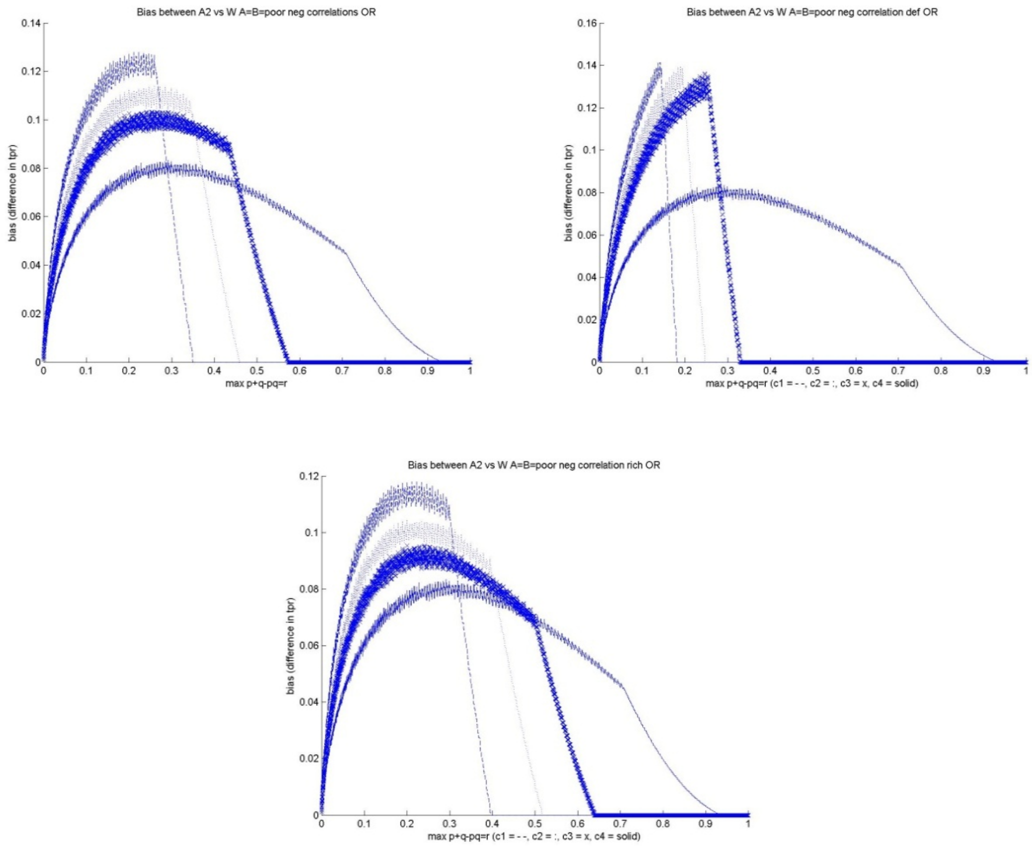



**Figure 5-29 – Altering the target environment when quantifying the performance bias between *across II* and *within* label-fused ROC curves under the *OR* rule.**

### *5.3.3 – Altered Target Environments Across III versus Within*

Given the similarities between *across I* and *across III* label fusion, the results of the simulation are predictable. The presence of the common target subset, $t_2$, reduces the bias between *across III* and *within* label-fused systems regardless of which Boolean rule is being utilized. Once again, all figures not discussed (negative correlation under the *AND* rule, positive correlation under the *OR* rule, and altering prior target prevalence for

individual classification systems under the *across III* label fusion rule) here are included in appendix B.

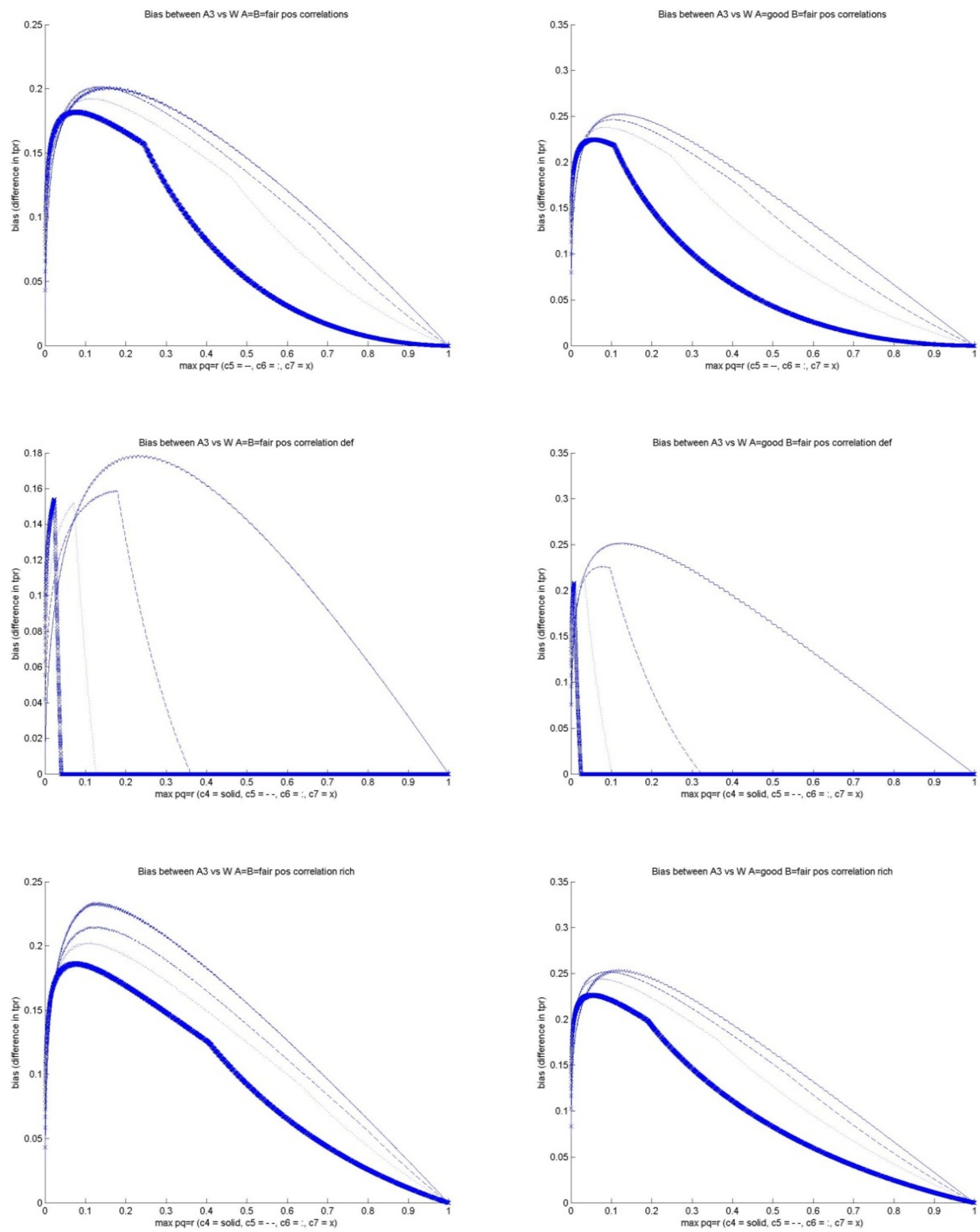



**Figure 5-30 – The presence of a shared subset seems to reduce the bias between *across III* and *within* ROC curves under the *AND* rule.**

In the case of the *OR* rule, the alteration of target environment produces similar results.

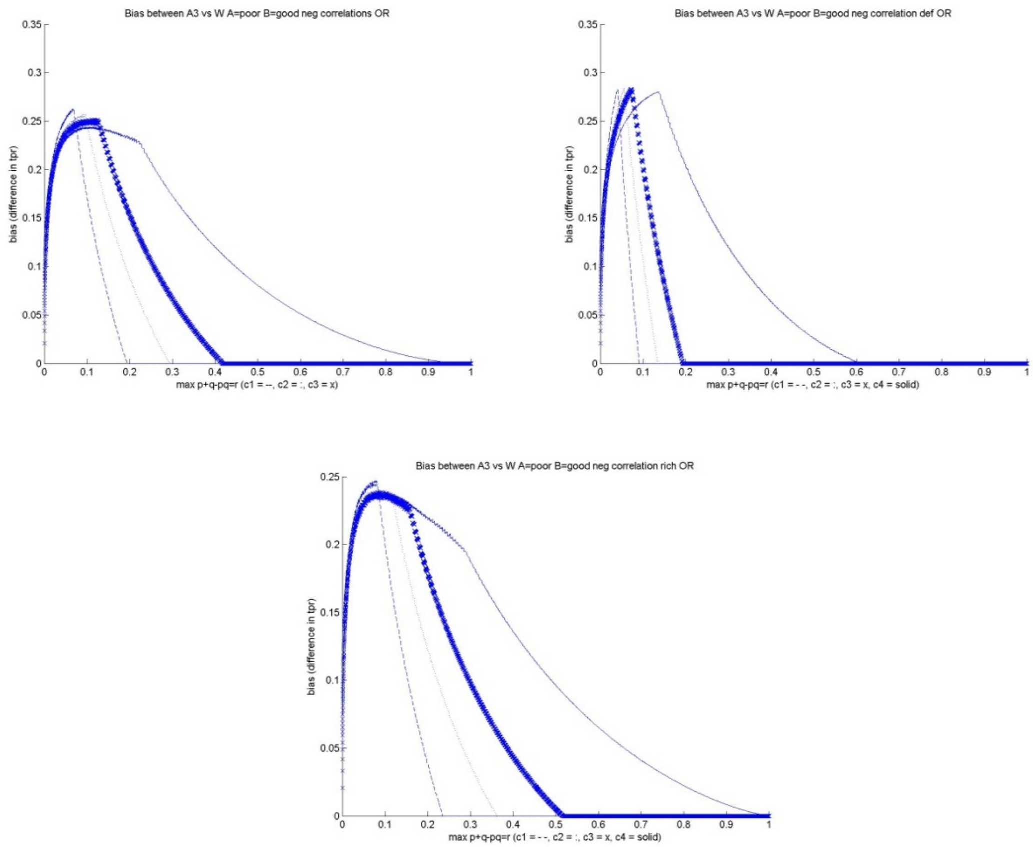



**Figure 5-31 – Altering target environment causes similar changes under the *OR* rule.**

In summary, all of the evidence points to a few clear things. Given the assumptions made in this simulation, the *within* combined classification system outperforms any *across* label-fused system in terms of pure performance. The *across* systems approaches the performance of the *within* system as the overlap in target classification grows. The performance bias can be tweaked as was demonstrated here. The most effective way to approximate these two ROC curves seems to be altering the probability of observing a target event either for *across* type targets or the proportion of targets in the population event set. Though clearly these cannot be changed in the field,

these are variable that the experimenter can use to alter the performance of legacy systems. Given that some trends were seen for the *AND* and *OR* rules, increasing or decreasing the correlation coefficient also gives the experimenter tools for adjusting the performance of legacy systems.

## *VI Discussion*

In this thesis, a method for determining the bias between different label-fused classification systems was developed. This notion of bias between label-fused classification systems carries its own set of pros and cons. First, the results of this thesis bring to light questions regarding the importance of pre-existing knowledge of those elements to be classified. Regardless of the type of Boolean rule, target environment, or prior target weighting, *within* label-fusion consistently outperforms any type of *across* label fusion. In the context of raw performance, it seems clear that one would choose to use *within* label-fusion. If the features of the elements of interest are not partitioned in such a way that the assumptions of *within* label fusion can be me, then there is the possibility that incorrectly applying this assumption overestimates system performance. As was demonstrated in the simulation, the difference in response between *within* and *across* label fusion could quite substantial and incorrectly applying the wrong type of fusion can lead to errors in reported performance.

That being said, there are ways to minimize the bias between the two systems. In *across I* and *across III* fused systems, if the prior target weight associated with the superior individual classification system (either system A or B) is increased, then the bias between *within* and *across* systems is decreased. Likewise, it was seen under the *AND* rule that *increasing* the correlation coefficient, $\rho$, can also assist in decreasing bias between the systems. Analogously, under the *OR* rule, decreasing $\rho$ reduced the bias between the two types of label-fused systems.

Recall that under most circumstances, target prevalence cannot be altered and pre-existing knowledge of the partition of the event set may be unknown. However, knowledge of a new target environment and the systems that are being fused(what the systems are tuned to detect), may provide the end-user with the means to appropriately adjust the performance and optimal thresholds for performance by fusing legacy systems. Such ability reflects a direct application of flexible engineering.

A limitation of using a set of empirical ROC curves is that a continuum of bias values will not always be available for computation (i.e. there may be values for the fused false positive "r" that are unique to each label-fused ROC curve, meaning not all bias values may be computed). Small departures from pure independence, that is, small values of correlation, may not alter the ROC curve of the fused system significantly. In this thesis, there were two cases where independence was crucial to theoretical and applicable development: 1) Individual classification systems $A_\theta$ and $B_\pi$ must be independent. 2) the classification systems must be conditionally independent with respect to the non-target partition of the event set. Much work has been done with respect to the first assumption and many researchers have been able to create independent classification systems. Furthermore, the cost functions derived by (*Schubert: 2005*) make no assumption about the independence of classification systems $A_\theta$ and $B_\pi$. Rather, the simplification of the cost function when the correlation constant $\rho = 0$ represents the formula when the two systems to be fused are independent. The second assumption is nested within the same equation; namely that the false positive for the label-fused system A and B under the *AND* rule is indeed pq (or p+q-pq under the *OR* rule) At times, this may be a strong assumption to make about the the classification systems and the non-target partition of

the event set. As a result, this assumption has to be built into any classification system that would use this approach. Future work may examine this independence and determine how sensitive the label-fused ROC curves are to this assumption.  It is not out of the realm of possibility that this assumption can be made, but to do so without closely considering the situation can lead to erroneous results.

Arguably the most important part of this thesis was the development of the formulas that quantify the performance bias between the different types of label-fusion. Given that these equations are built from the cost functions themselves, they include all the variables necessary to make the transformation from one label-fused ROC curve to another other. Secondly, these equations are built from the individual classification systems $A_\theta$ and $B_\pi$; meaning this is ample information to fuse the classification systems and determine the difference in performance between them. Hence, given knowledge about systems $A_\theta$ and $B_\pi$, it is possible to pinpoint those variables that are causing the bias and alter them accordingly. It is assumed that if label-fusion is going to be used, one has to make an assumption regarding the partitioning of the event set. Regardless of what this original assumption may be, as more knowledge becomes available about the true partitions, these variables can be tuned to model the dynamic truth. In such a manner, determining the performance of the combined classification system is no longer a barrier when developing such flexibility in system design.

In order to use these formulas, a hard value for the prior probability of target and non-target events is needed. Choosing only one of a continuum of values may introduce errors, but the use of training (such as with neural networks) may mitigate these potential issues.

Should values be available, one benefit of having the simple formulas presented herein is that it is very easy to apply computationally. The algorithm provided in appendix C requires only finding the set intersection of false positive values and subtracting the corresponding true positive values to compute the bias and adjust performance between *within* and *across* label-fused ROC curves. Another benefit associated with these formulas is that they can be easily applied to legacy data. Given the tenuous future of research funding, the ability to apply new ideas to pre-existing scientific data (some of which have already made assumptions regarding the distribution of target and non-target features), these functions will be easily adaptable. Further, these formulas also support system flexibility so that different combinations of individual classification systems may be combined together and performance appropriately determined from only the information of the performances (ROC curves) of the individual systems.

### *6.1 Future Work*

It is clear that there are plenty of questions regarding this approach to fusion that have yet to be answered. First, the development of a distribution for the prior probability of targets and non-targets appears pressing. In this way, the weighting associated with individual target types can be adjusted dynamically. Assuming values for the prior probability of targets and non-targets in an environment is perfectly suited to theory, but in practice, building a system this specific is highly inefficient. Second, developing a method that eases the need for conditional independence of classification systems with respect to the non-target event set could have interesting ramifications. It is common in research to suggest that theoretical classification systems are indeed "independent," but little work has been done showing what effects this has in application. Finally, it is of

interest to extend the concepts developed here to any type of ROC curve. Remember that the bias formulas currently only work for the Boolean *AND* and Boolean *OR* rule. It would be interesting to extend this theory to different classification methods such as neural networks or at least different performance measures.

*Appendix A*

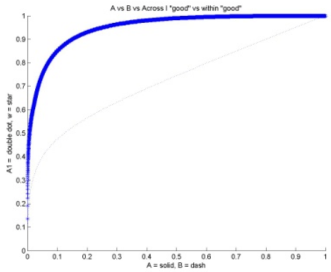
In this appendix you will find all of the ROC curves that were created during the simulation. As a reference, in each figure, the ROC curves of the specified combined *across* and *within* classification system are plotted along with the ROC curves for individual systems $A_\theta$ and $B_{\gamma,\varepsilon}$. The material will be presented in the following order:

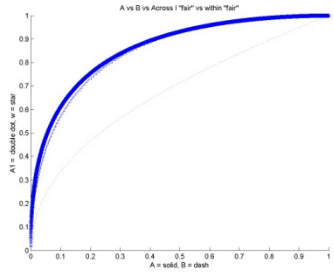
1) Combined systems in the *enhanced target prevalence* environment
   a) Combined *AND* ROC curves
   b) Combined *OR* ROC curves
   c) Combined *AND* ROC curves with unequal prior target weighting
   d) Combined *OR* ROC curves with unequal prior target weighting

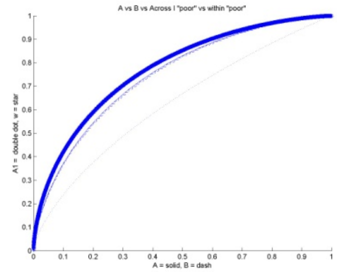
2) Combined systems in the *rich target prevalence* environment
   a) Combined *AND* ROC curves
   b) Combined *OR* ROC curves
   c) Combined *AND* ROC curves with unequal prior target weighting
   d) Combined *OR* ROC curves with unequal prior target weighting

3) Combined systems in the *deficient target prevalence* environment
   a) Combined *AND* ROC curves
   b) Combined *OR* ROC curves
   c) Combined *AND* ROC curves with unequal prior target weighting
   d) Combined *OR* ROC curves with unequal prior target weighting

## *A.1 Enhanced Target Prevalence Figures*



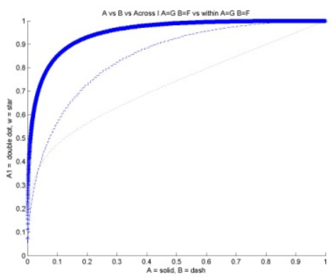



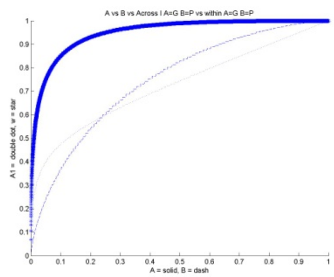



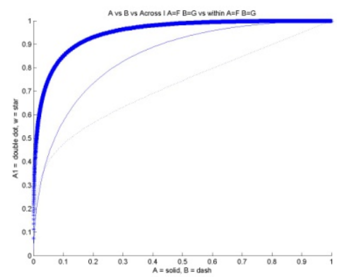

**A=B=*good AND* A1 vs Within**

**A=B=*fair AND* A1 vs Within**

**A=B=*poor AND* A1 vs Within**



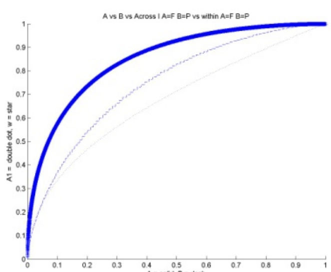



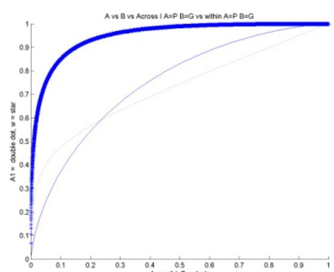



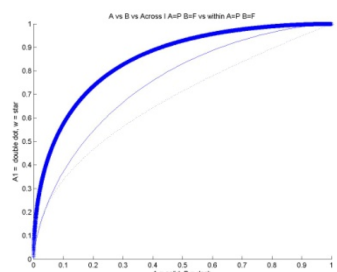

**A=*good* B=*fair AND* A1 vs Within**

**A=*good* B=*poor AND* A1 vs Within**

**A=*fair* B=*good AND* A1 vs Within**



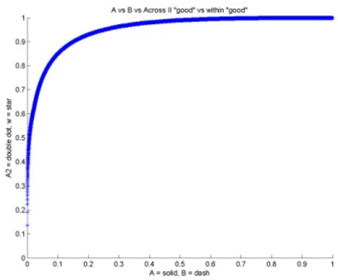



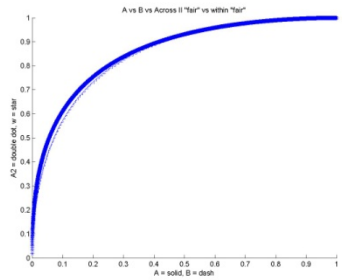

**A=*fair* B=*poor AND* A1 vs Within**

**A=*poor* B=*good AND* A1 vs Within**

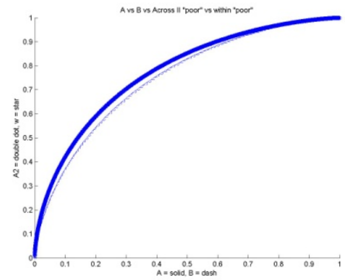
**A=*poor* B=*fair AND* A1 vs Within**



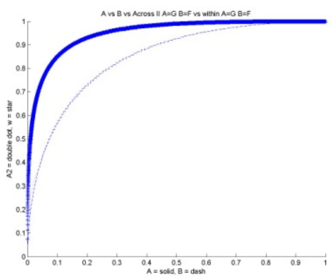

**A=B=*good AND* A2 vs Within**



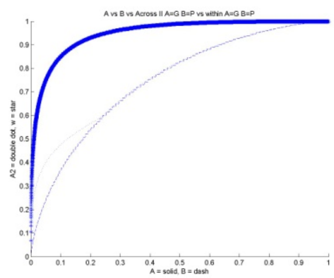

**A=B=*fair AND* A2 vs Within**



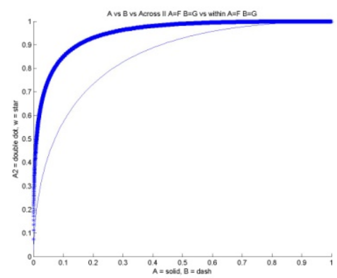

**A=B=*poor AND* A2 vs Within**



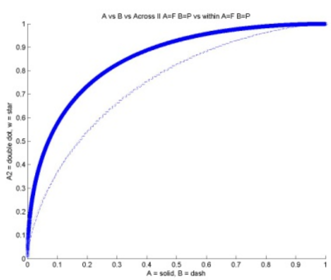

**A=*good* B=*fair AND* A2 vs Within**



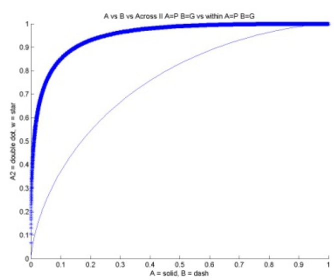

**A=*good* B=*poor AND* A2 vs Within**



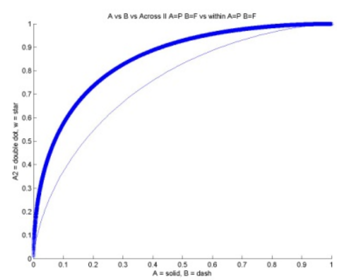

**A=*fair* B=*good AND* A2 vs Within**



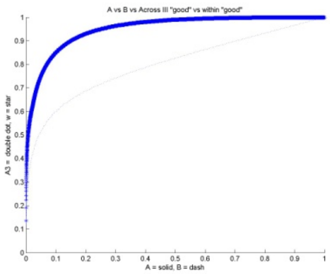

**A=*fair* B=*poor AND* A2 vs Within**

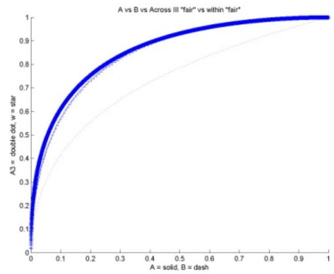


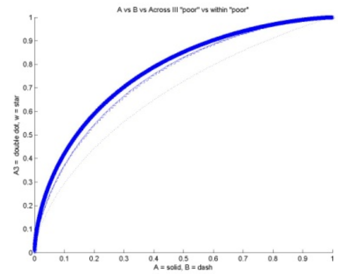

**A=*poor* B=*good AND* A2 vs Within**




**A=*poor* B=*fair AND* A2 vs Within**



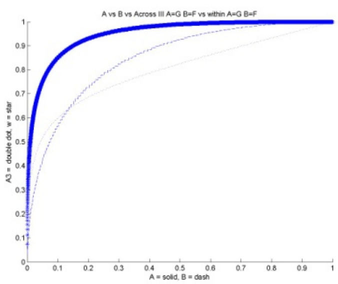



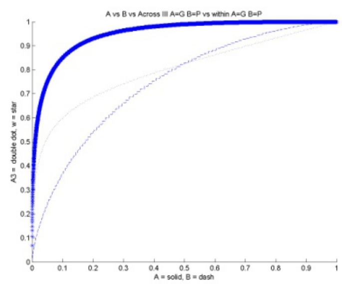



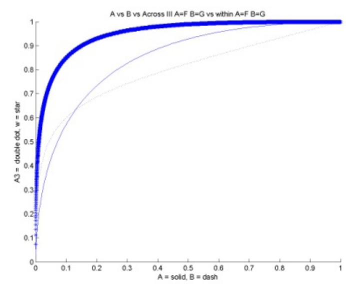

**A=B=*good AND* A3 vs Within**

**A=B=*fair AND* A3 vs Within**

**A=B=*poor AND* A3 vs Within**



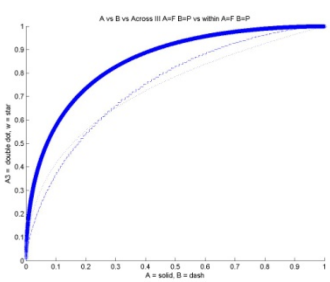



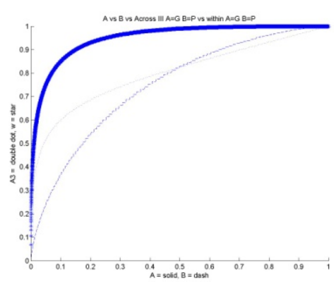



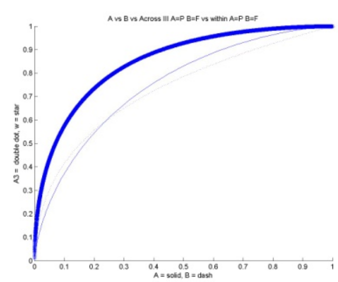

**A=*good* B=*fair AND* A3 vs Within**

**A=*good* B=*poor AND* A3 vs Within**

**A=*fair* B=*good AND* A3 vs Within**



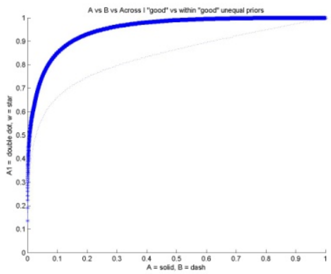



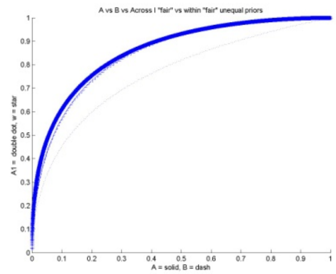

**A=*fair* B=*poor AND* A3 vs Within**

**A=*poor* B=*good AND* A3 vs Within**

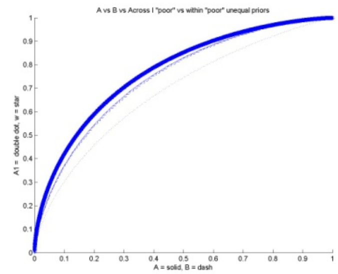
**A=*poor* B=*fair AND* A3 vs Within**



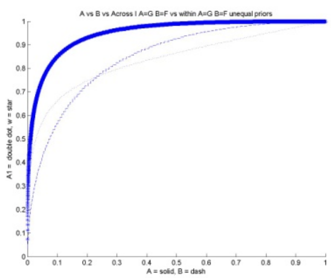

**A=B=*good AND* unequal priors A1 vs Within**



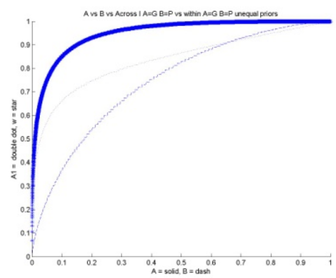

**A=B=*fair AND* unequal priors A1 vs Within**



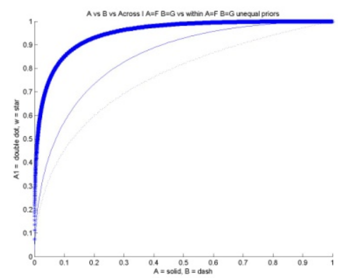

**A=B=*poor AND* unequal priors A1 vs Within**



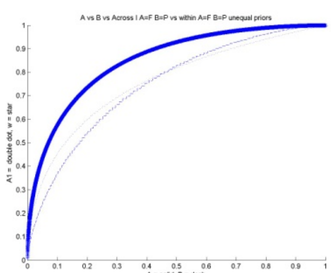

**A=*good* B=*fair AND* unequal priors A1 vs Within**



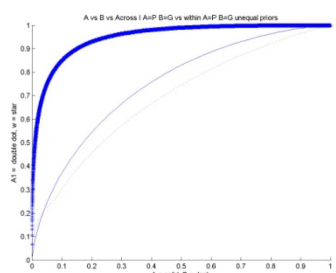

**A=*good* B=*poor AND* unequal priors A1 vs Within**



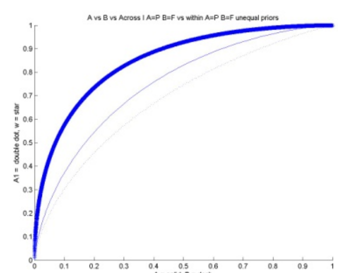

**A=*fair* B=*good AND* unequal priors A1 vs Within**



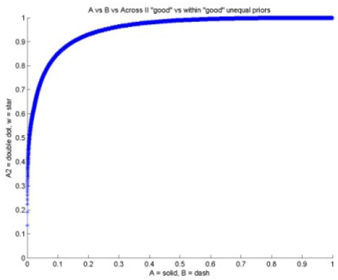

**A=*fair* B=*poor AND* unequal priors A1 vs Within**



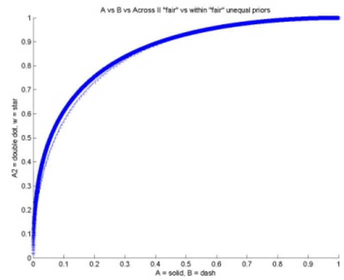

**A=*poor* B=*good AND* unequal priors A1 vs Within**



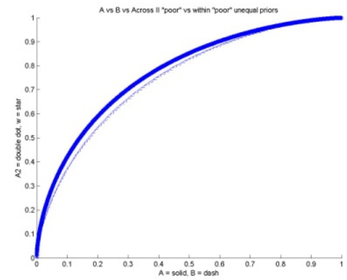

**A=*poor* B=*fair AND* unequal priors A1 vs Within**



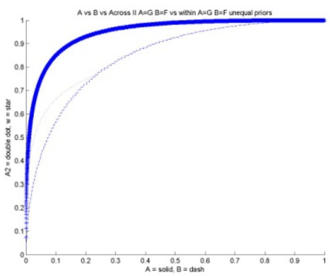

**A=B=*good AND* unequal priors A2 vs Within**



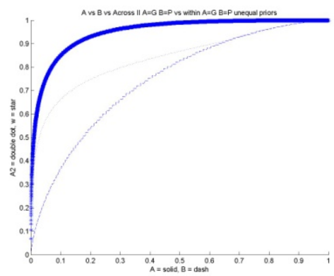

**A=B=*fair AND* unequal priors A2 vs Within**



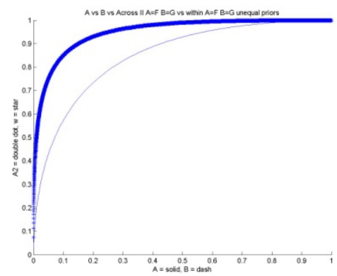

**A=B=*poor AND* unequal priors A2 vs Within**



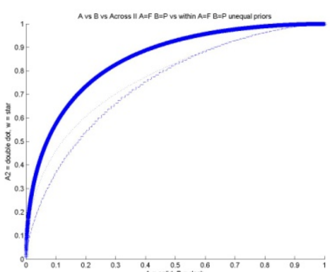

**A=*good* B=*fair AND* unequal priors A2 vs Within**



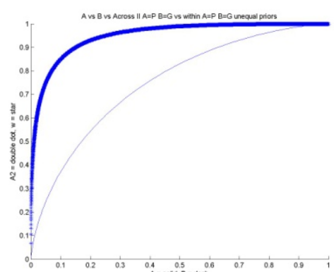

**A=*good* B=*poor AND* unequal priors A2 vs Within**



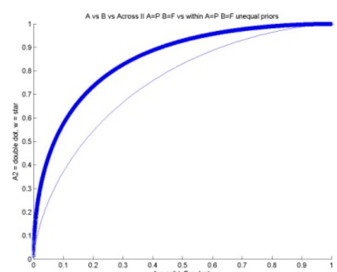

**A=*fair* B=*good AND* unequal priors A2 vs Within**



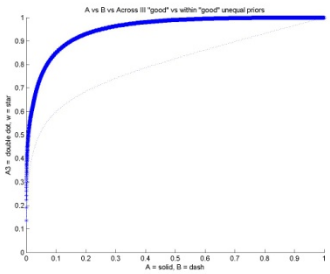

**A=*fair* B=*poor AND* unequal priors A2 vs Within**



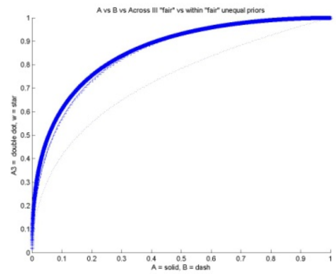

**A=*poor* B=*good AND* unequal priors A2 vs Within**



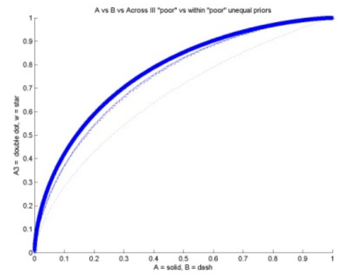

**A=*poor* B=*fair AND* unequal priors A2 vs Within**



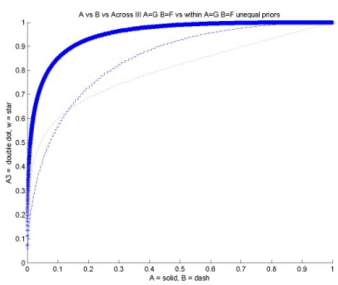

**A=B=*good AND* unequal priors A3 vs Within**



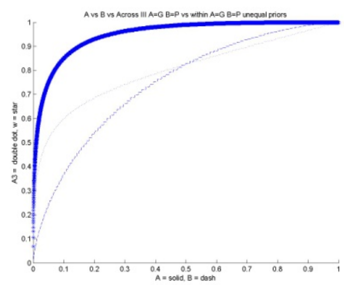

**A=B=*fair AND* unequal priors A3 vs Within**



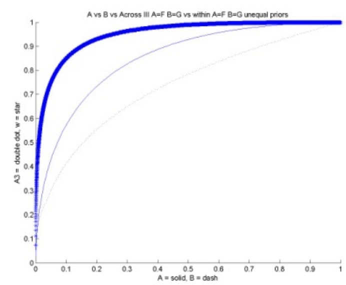

**A=B=*poor AND* unequal priors A3 vs Within**



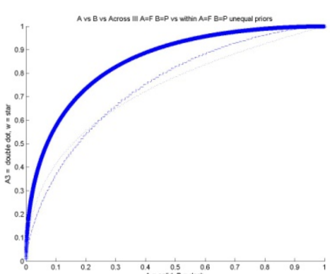

**A=*good* B=*fair AND* unequal priors A3 vs Within**



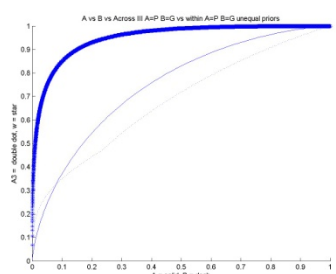

**A=*good* B=*poor AND* unequal priors A3 vs Within**



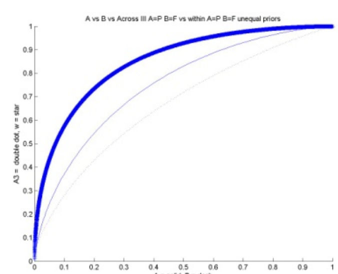

**A=*fair* B=*good AND* unequal priors A3 vs Within**



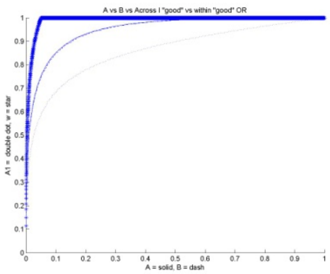

**A=*fair* B=*poor AND* unequal priors A3 vs Within**



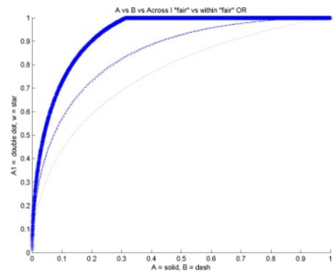

**A=*poor* B=*good AND* unequal priors A3 vs Within**



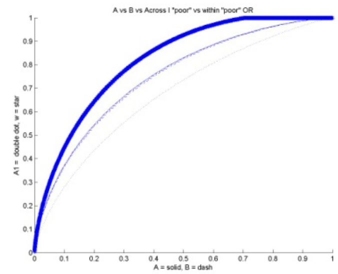

**A=*poor* B=*fair AND* unequal priors A3 vs Within**



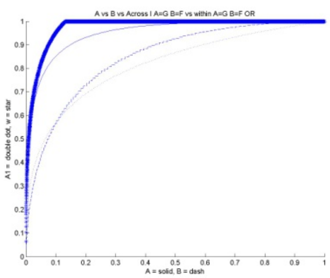

**A=B=*good OR* A1 vs Within**



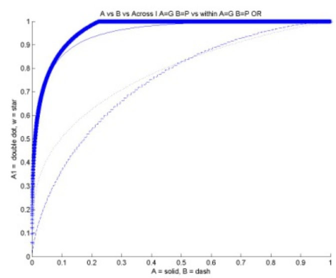

**A=B=*fair OR* A1 vs Within**



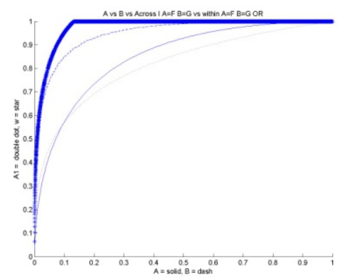

**A=B=*poor OR* A1 vs Within**



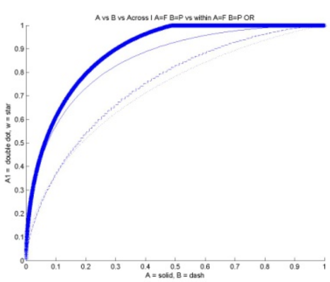

**A=*good* B=*fair OR* A1 vs Within**



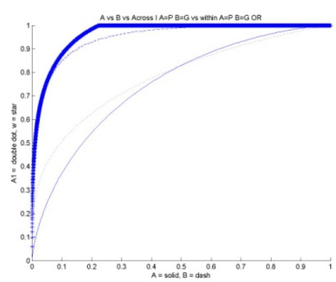

**A=*good* B=*poor OR* A1 vs Within**



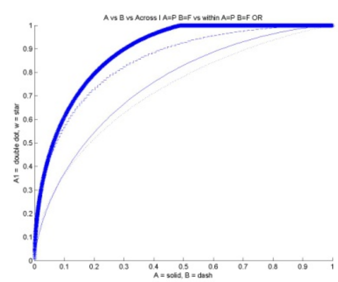

**A=*fair* B=*good OR* A1 vs Within**



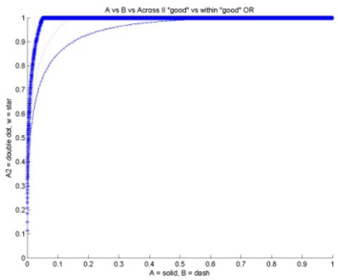

**A=*fair* B=*poor OR* A1 vs Within**



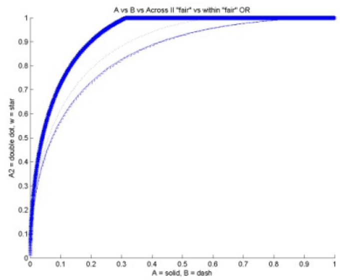

**A=*poor* B=*good OR* A1 vs Within**



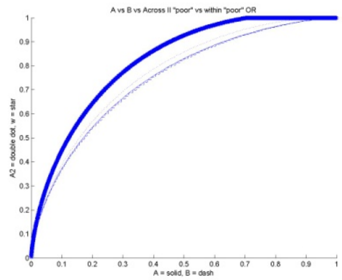

**A=*poor* B=*fair OR* A1 vs Within**



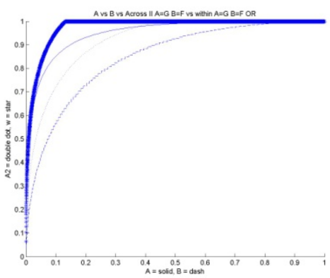

**A=B=*good OR* A2 vs Within**



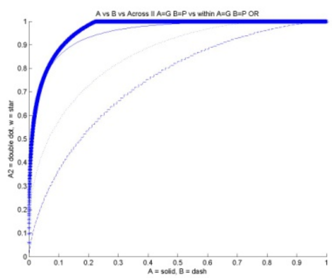

**A=B=*fair OR* A2 vs Within**



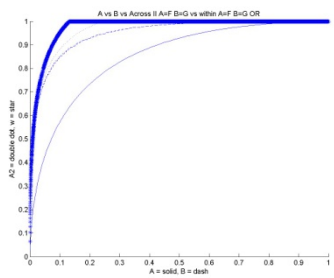

**A=B=*poor OR* A2 vs Within**



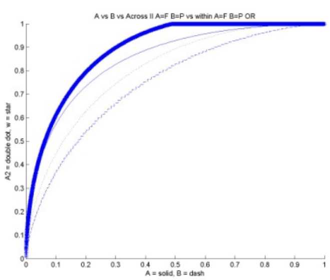

**A=*good* B=*fair OR* A2 vs Within**



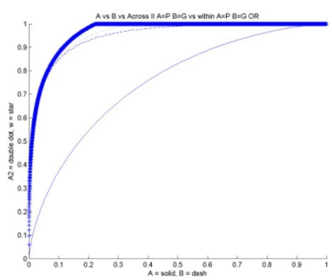

**A=*good* B=*poor OR* A2 vs Within**



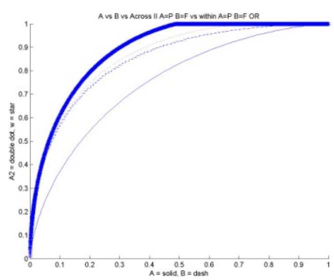

**A=*fair* B=*good OR* A2 vs Within**



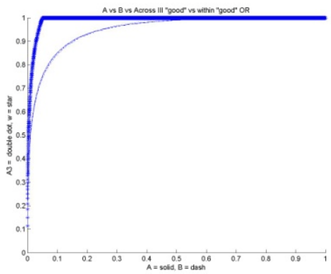

**A=*fair* B=*poor OR* A2 vs Within**



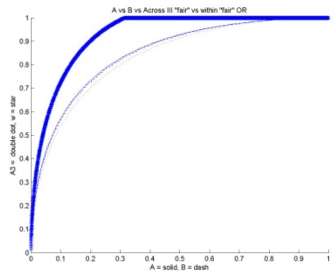

**A=*poor* B=*good OR* A2 vs Within**



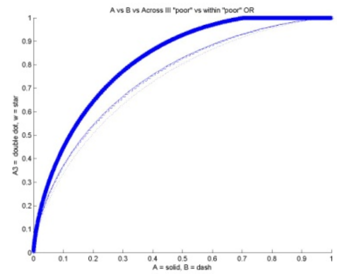

**A=*poor* B=*fair OR* A2 vs Within**



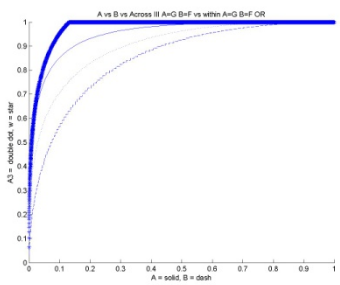

**A=B=*good OR* A3 vs Within**



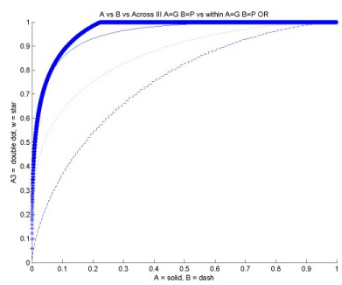

**A=B=*fair OR* A3 vs Within**



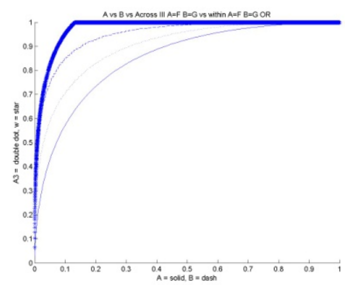

**A=B=*poor OR* A3 vs Within**



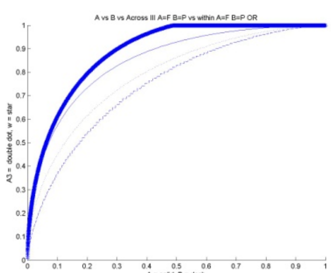

**A=*good* B=*fair OR* A3 vs Within**



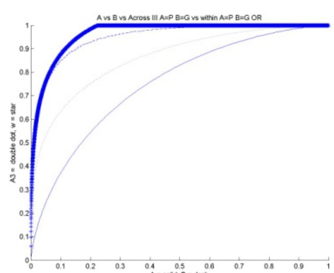

**A=*good* B=*poor OR* A3 vs Within**



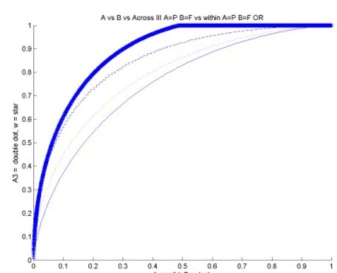

**A=*fair* B=*good OR* A3 vs Within**



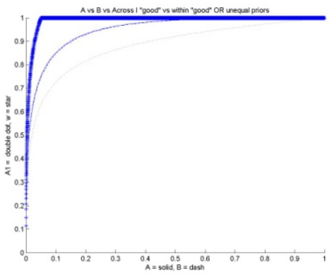

**A=*fair* B=*poor OR* A3 vs Within**



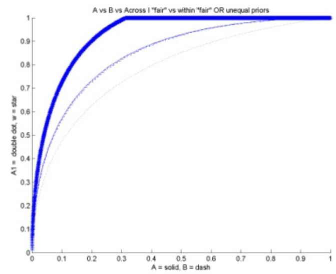

**A=*poor* B=*good OR* A3 vs Within**



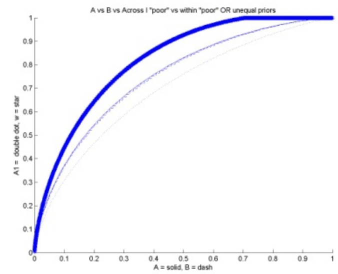

**A=*poor* B=*fair OR* A3 vs Within**



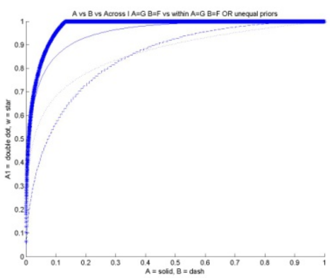

**A=B=*good OR* unequal priors A1 vs Within**



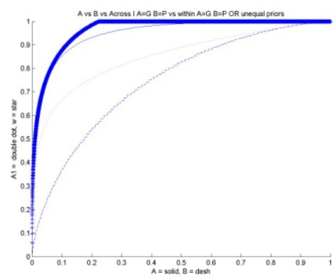

**A=B=*fair OR* unequal priors A1 vs Within**



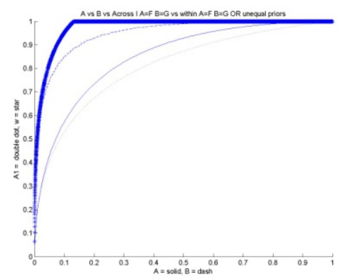

**A=B=*poor OR* unequal priors A1 vs Within**



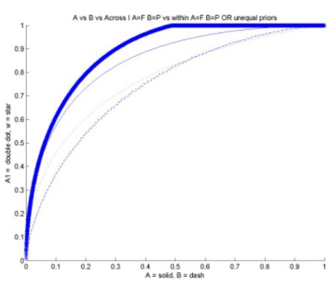

**A=*good* B=*fair OR* unequal priors A1 vs Within**



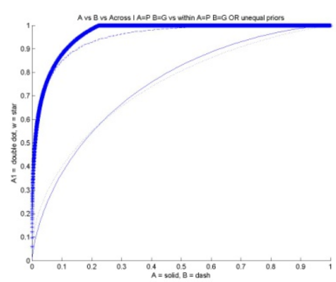

**A=*good* B=*poor OR* unequal priors A1 vs Within**



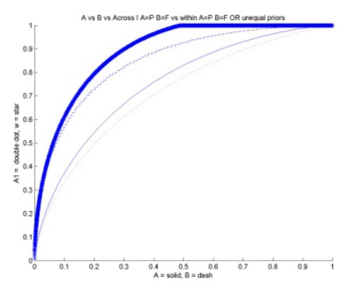

**A=*fair* B=*good OR* unequal priors A1 vs Within**



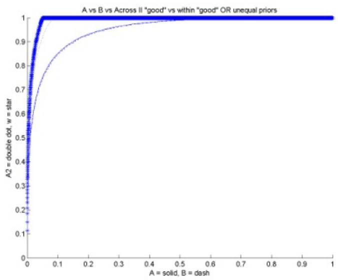

**A=*fair* B=*poor OR* unequal priors A1 vs Within**



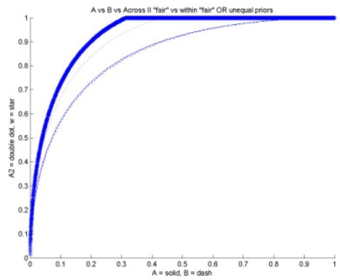

**A=*poor* B=*good OR* unequal priors A1 vs Within**



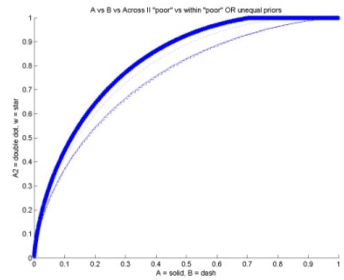

**A=*poor* B=*fair OR* unequal priors A1 vs Within**



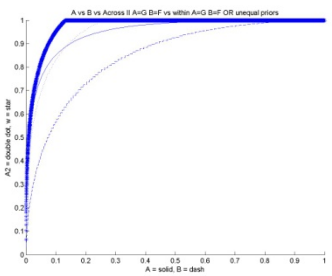

**A=B=*good OR* unequal priors A2 vs Within**



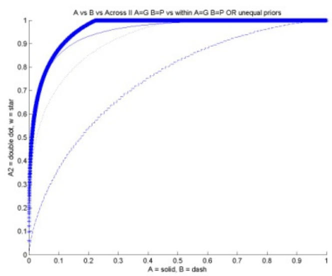

**A=B=*fair OR* unequal priors A2 vs Within**



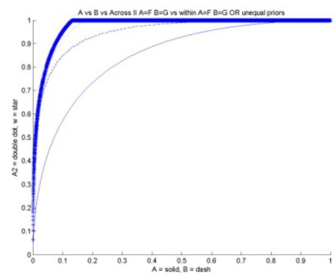

**A=B=*poor OR* unequal priors A2 vs Within**



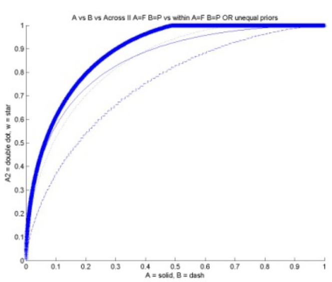

**A=*good* B=*fair OR* unequal priors A2 vs Within**



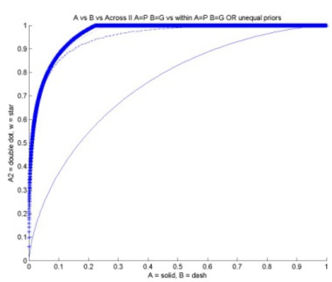

**A=*good* B=*poor OR* unequal priors A2 vs Within**



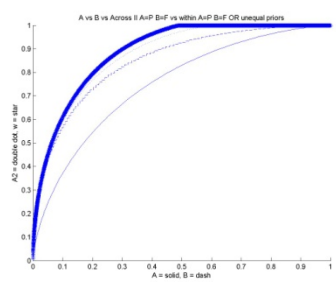

**A=*fair* B=*good OR* unequal priors A2 vs Within**



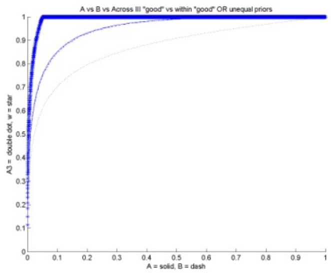

**A=*fair* B=*poor OR* unequal priors A2 vs Within**



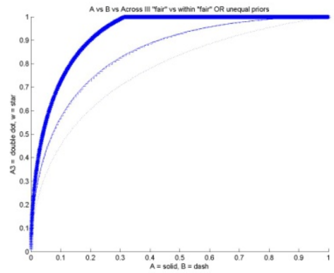

**A=*poor* B=*good OR* unequal priors A2 vs Within**



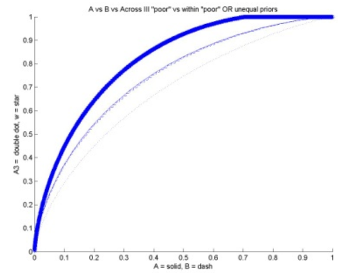

**A=*poor* B=*fair OR* unequal priors A2 vs Within**



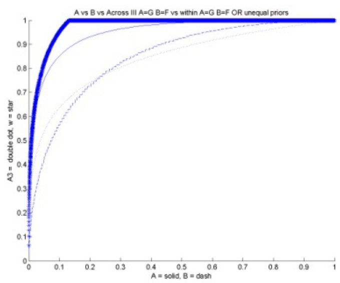

**A=B=*good OR* unequal priors A3 vs Within**



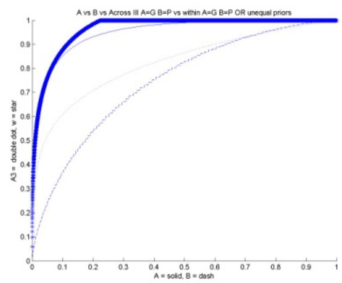

**A=B=*fair OR* unequal priors A3 vs Within**



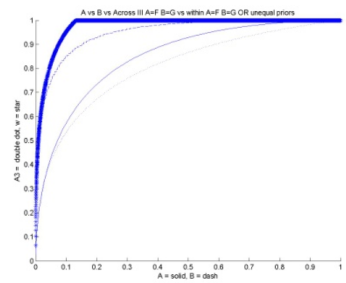

**A=B=*poor OR* unequal priors A3 vs Within**



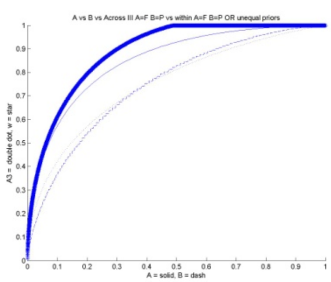

**A=*good* B=*fair OR* unequal priors A3 vs Within**



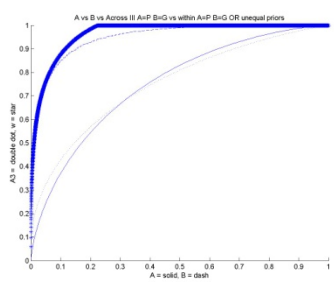

**A=*good* B=*poor OR* unequal priors A3 vs Within**



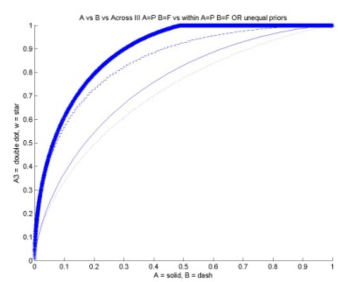

**A=*fair* B=*good OR* unequal priors A3 vs Within**




**A=*fair* B=*poor OR* unequal priors A3 vs Within**

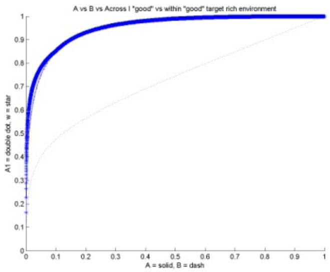



**A=*poor* B=*good OR* unequal priors A3 vs Within**

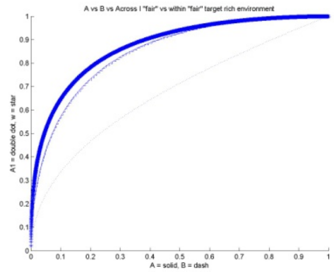



**A=*poor* B=*fair OR* unequal priors A3 vs Within**

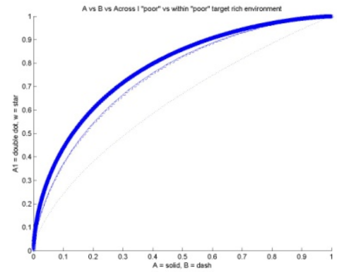
## *A.2 – Rich Target Prevalence Figures*



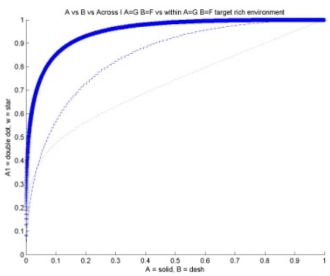

**A=B=*good AND* A1 vs Within**



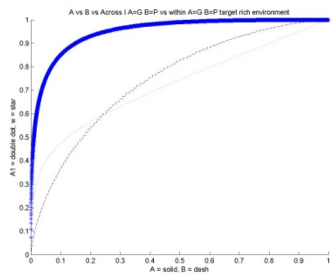

**A=B=*fair AND* A1 vs Within**



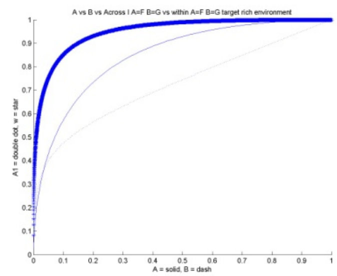

**A=B=*poor AND* A1 vs Within**



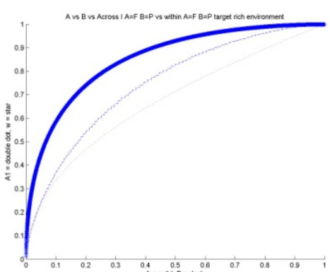

**A=*good* B=*fair AND* A1 vs Within**



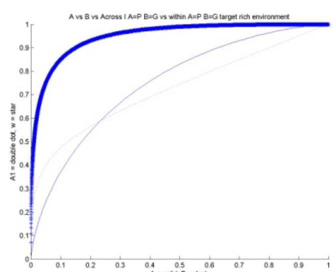

**A=*good* B=*poor AND* A1 vs Within**



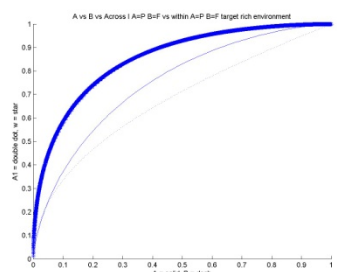

**A=*fair* B=*good AND* A1 vs Within**



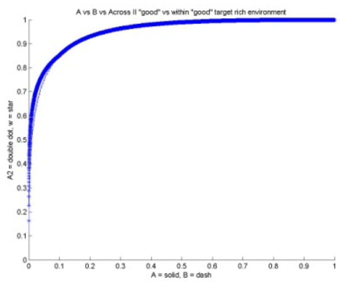

**A=*fair* B=*poor AND* A1 vs Within**



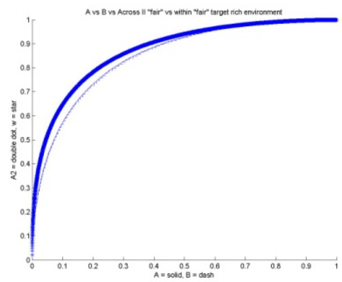

**A=*poor* B=*good AND* A1 vs Within**



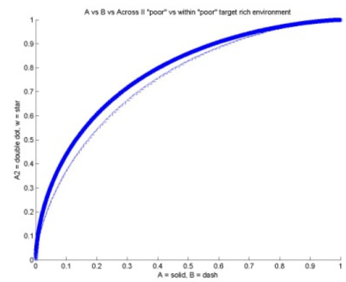

**A=*poor* B=*fair AND* A1 vs Within**



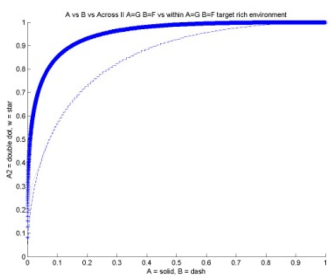

**A=B=*good AND* A2 vs Within**



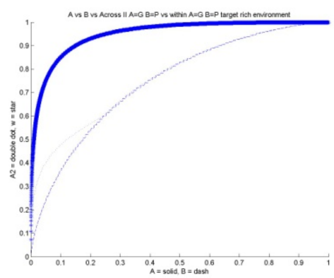

**A=B=*fair AND* A2 vs Within**



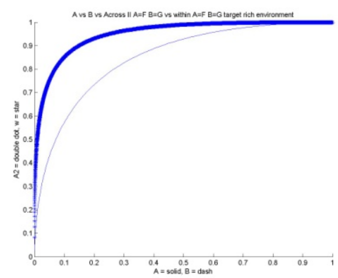

**A=B=*poor AND* A2 vs Within**



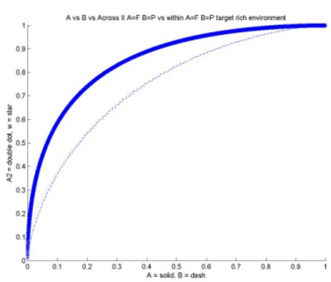

**A=*good* B=*fair AND* A2 vs Within**



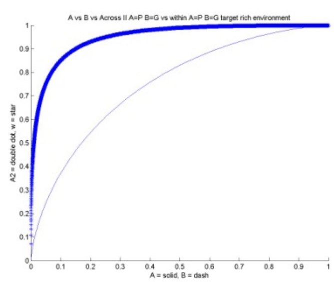

**A=*good* B=*poor AND* A2 vs Within**



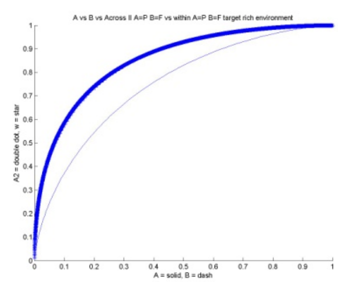

**A=*fair* B=*good AND* A2 vs Within**



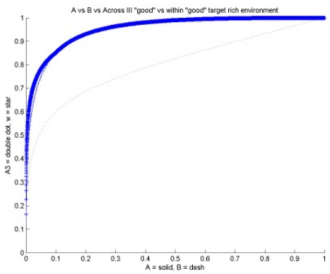

**A=*fair* B=*poor AND* A2 vs Within**



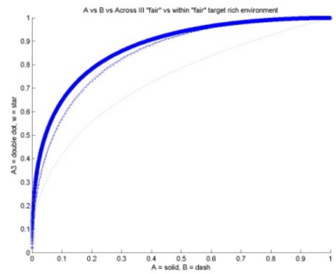

**A=*poor* B=*good AND* A2 vs Within**



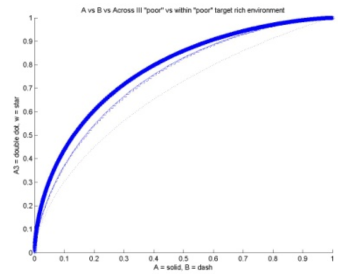

**A=*poor* B=*fair AND* A2 vs Within**



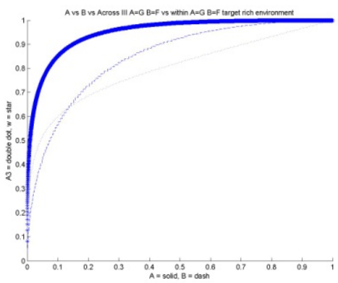

**A=B=*good AND* A3 vs Within**



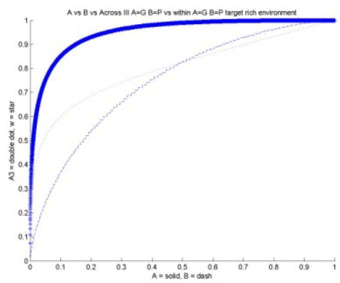

**A=B=*fair AND* A3 vs Within**



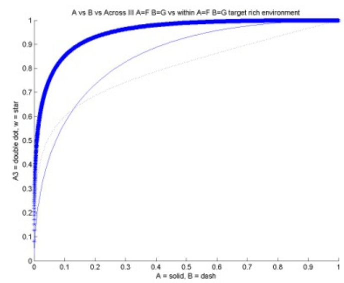

**A=B=*poor AND* A3 vs Within**



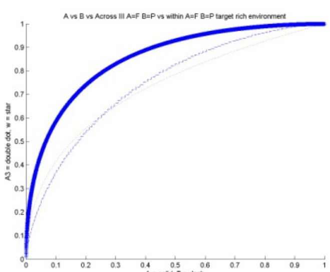

**A=*good* B=*fair AND* A3 vs Within**



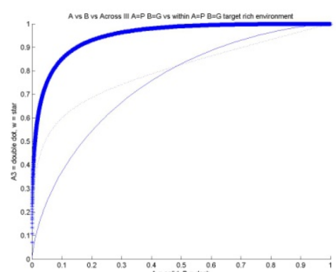

**A=*good* B=*poor AND* A3 vs Within**



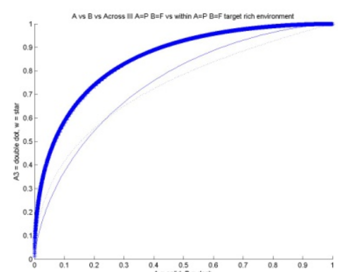

**A=*fair* B=*good AND* A3 vs Within**



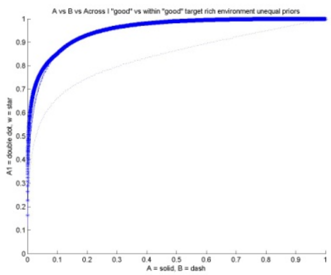

**A=*fair* B=*poor AND* A3 vs Within**



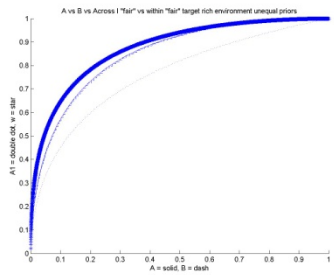

**A=*poor* B=*good AND* A3 vs Within**



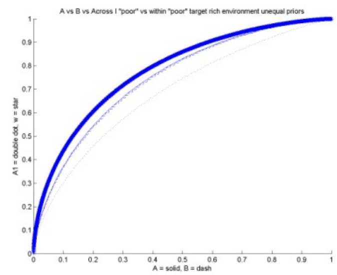

**A=*poor* B=*fair AND* A3 vs Within**



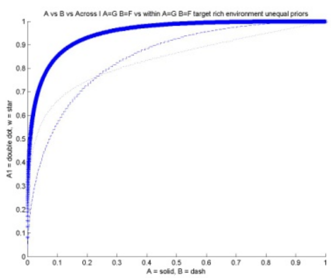

**A=B=*good AND* unequal priors A1 vs Within**



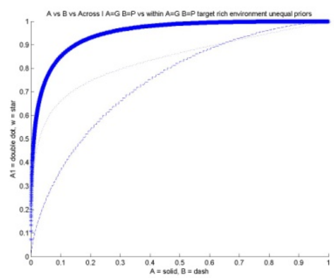

**A=B=*fair AND* unequal priors A1 vs Within**



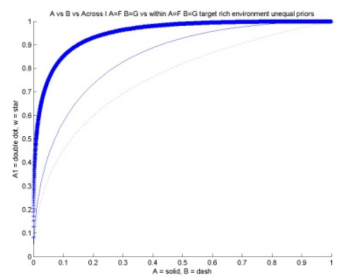

**A=B=*poor AND* unequal priors A1 vs Within**



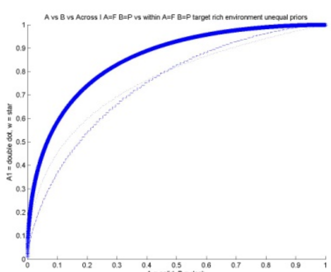

**A=*good* B=*fair AND* unequal priors A1 vs Within**



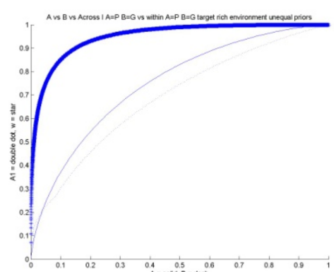

**A=*good* B=*poor AND* unequal priors A1 vs Within**



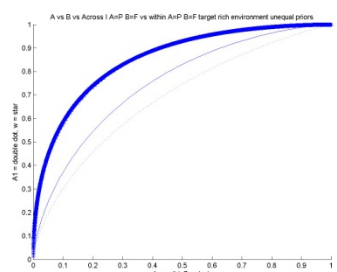

**A=*fair* B=*good AND* unequal priors A1 vs Within**



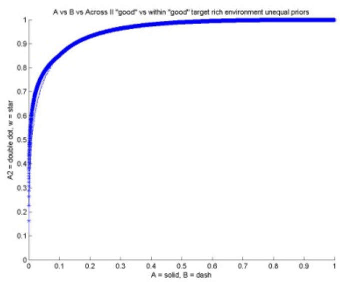

**A=*fair* B=*poor AND* unequal priors A1 vs Within**



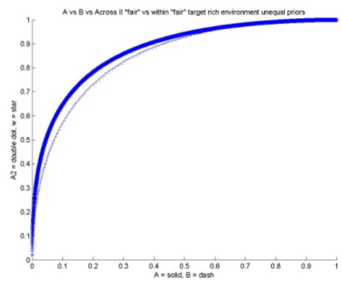

**A=*poor* B=*good AND* unequal priors A1 vs Within**



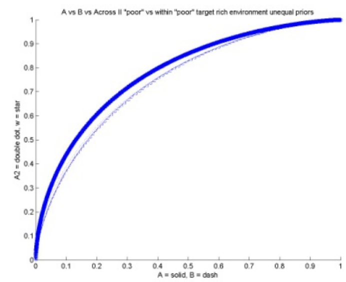

**A=*poor* B=*fair AND* unequal priors A1 vs Within**



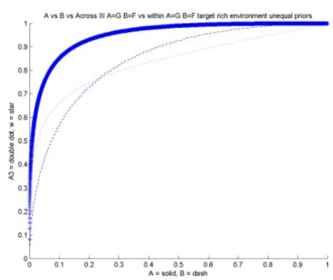
**A=B=*good AND* unequal priors A2 vs Within**



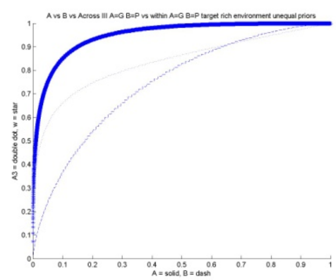
**A=B=*fair AND* unequal priors A2 vs Within**



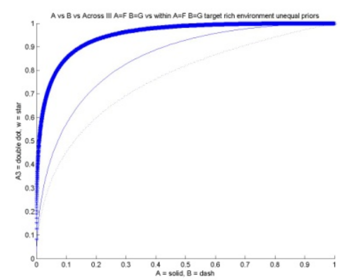
**A=B=*poor AND* unequal priors A2 vs Within**



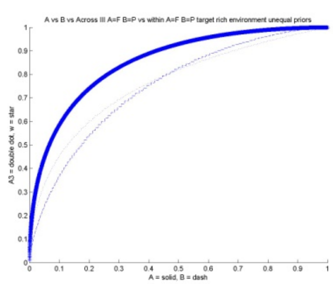
**A=*good* B=*fair AND* unequal priors A2 vs Within**



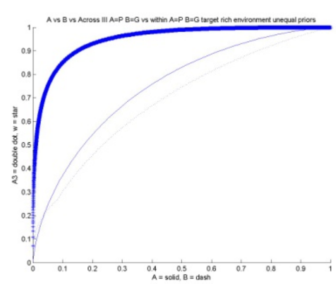
**A=*good* B=*poor AND* unequal priors A2 vs Within**



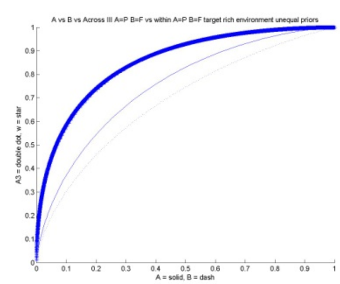
**A=*fair* B=*good AND* unequal priors A2 vs Within**



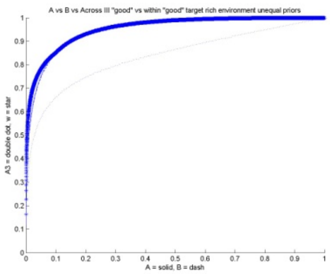
**A=*fair* B=*poor AND* unequal priors A2 vs Within**



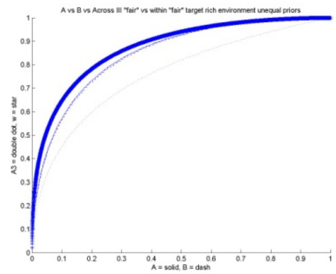
**A=*poor* B=*good AND* unequal priors A2 vs Within**



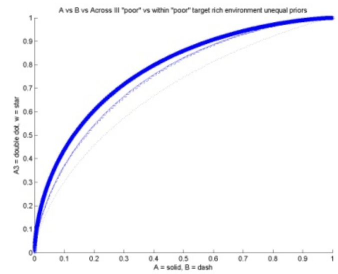
**A=*poor* B=*fair AND* unequal priors A2 vs Within**



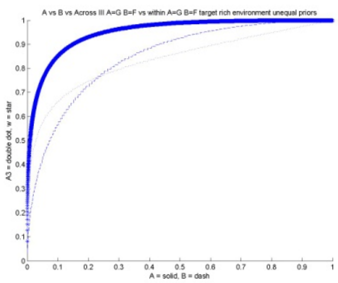

**A=B=*good AND* unequal priors A3 vs Within**



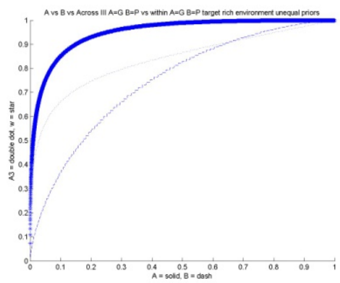

**A=B=*fair AND* unequal priors A3 vs Within**



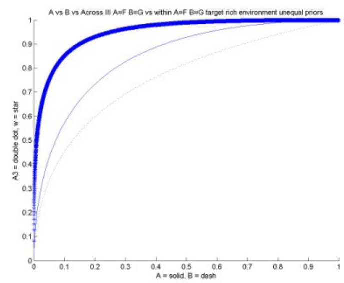

**A=B=*poor AND* unequal priors A3 vs Within**



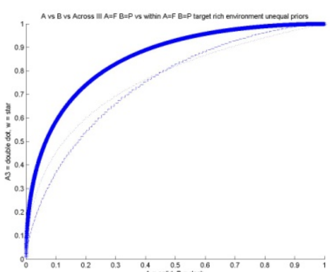

**A=*good* B=*fair AND* unequal priors A3 vs Within**



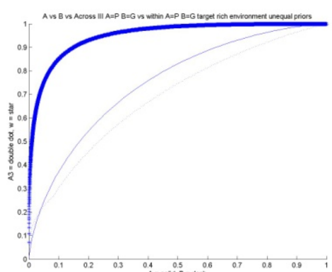

**A=*good* B=*poor AND* unequal priors A3 vs Within**



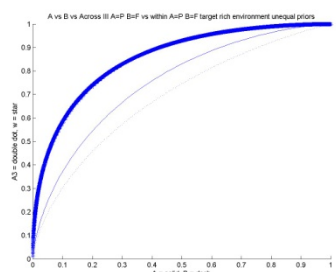

**A=*fair* B=*good AND* unequal priors A3 vs Within**



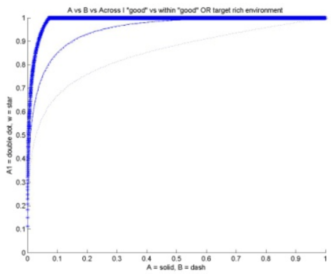

**A=*fair* B=*poor AND* unequal priors A3 vs Within**



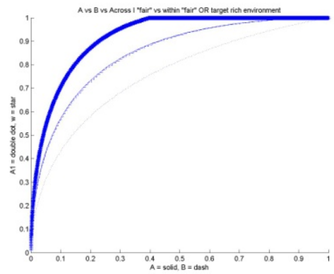

**A=*poor* B=*good AND* unequal priors A3 vs Within**



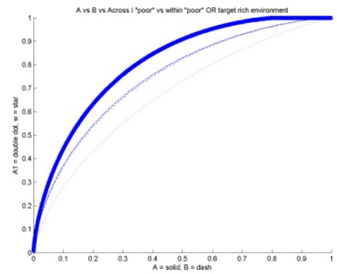

**A=*poor* B=*fair AND* unequal priors A3 vs Within**



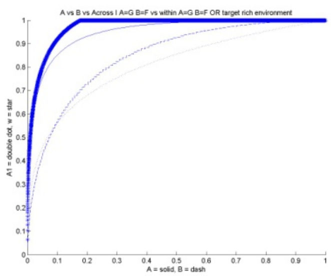

**A=B=*good OR* A1 vs Within**



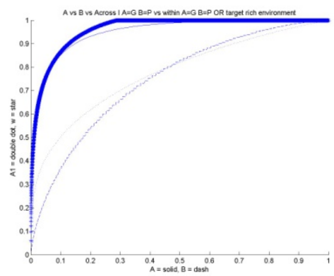

**A=B=*fair OR* A1 vs Within**



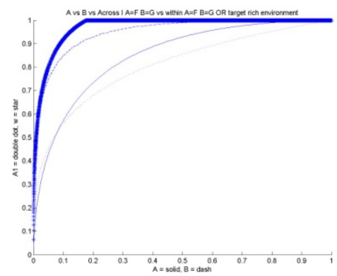

**A=B=*poor OR* A1 vs Within**



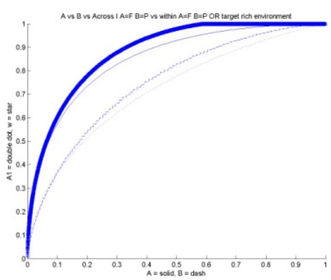

**A=*good* B=*fair OR* A1 vs Within**



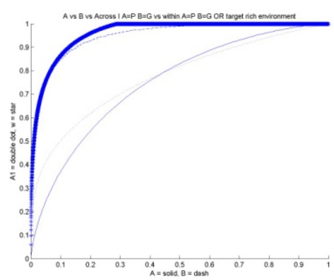

**A=*good* B=*poor OR* A1 vs Within**



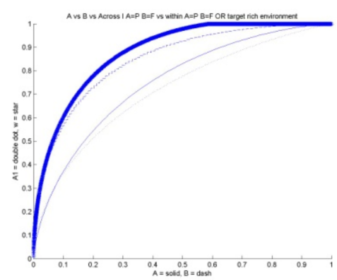

**A=*fair* B=*good OR* A1 vs Within**



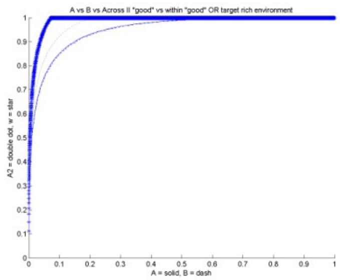

**A=*fair* B=*poor OR* A1 vs Within**



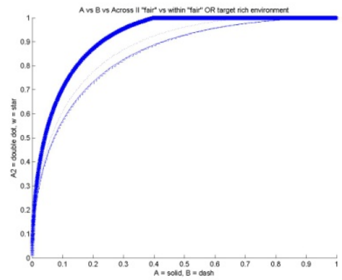

**A=*poor* B=*good OR* A1 vs Within**



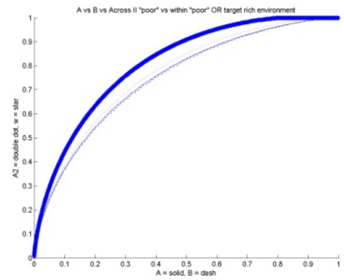

**A=*poor* B=*fair OR* A1 vs Within**



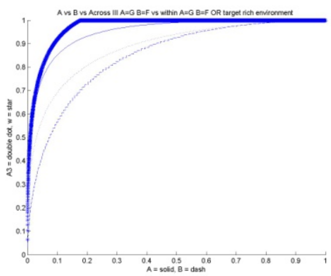

**A=B=*good OR* A2 vs Within**



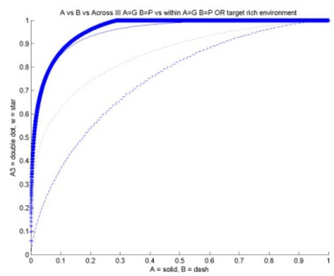

**A=B=*fair OR* A2 vs Within**



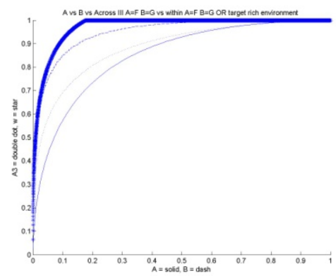

**A=B=*poor OR* A2 vs Within**



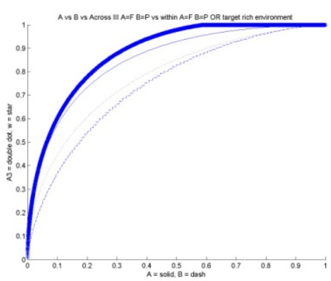

**A=*good* B=*fair OR* A2 vs Within**



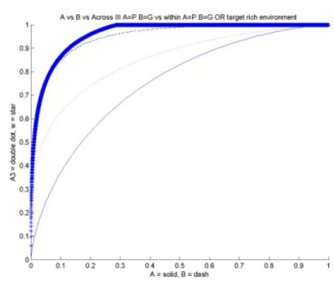

**A=*good* B=*poor OR* A2 vs Within**



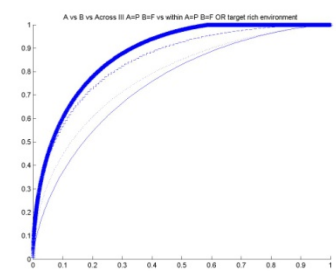

**A=*fair* B=*good OR* A2 vs Within**



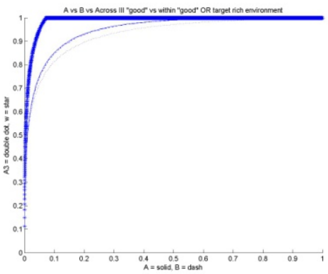

**A=*fair* B=*poor OR* A2 vs Within**



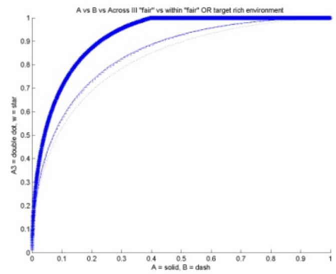

**A=*poor* B=*good OR* A2 vs Within**



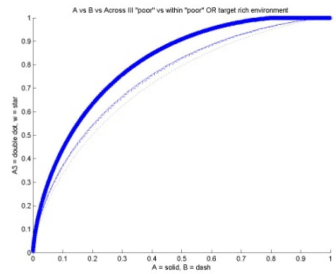

**A=*poor* B=*fair OR* A2 vs Within**



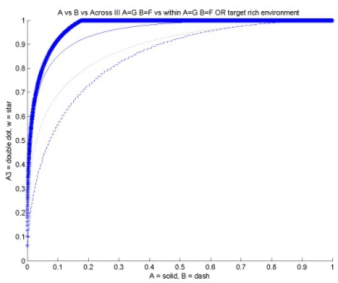

**A=B=*good OR* A3 vs Within**



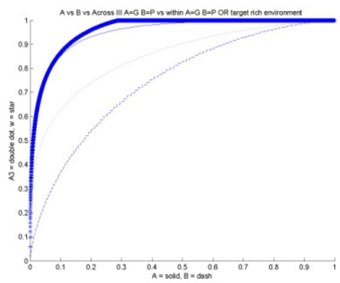

**A=B=*fair OR* A3 vs Within**



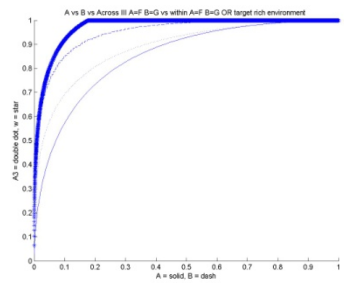

**A=B=*poor OR* A3 vs Within**



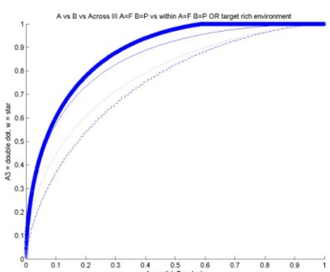

**A=*good* B=*fair OR* A3 vs Within**



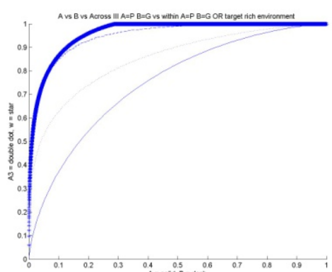

**A=*good* B=*poor OR* A3 vs Within**



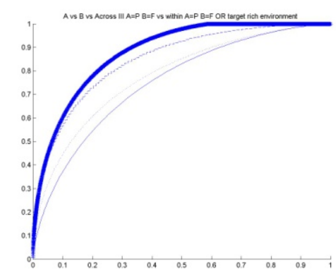

**A=*fair* B=*good OR* A3 vs Within**



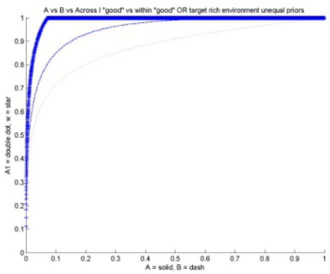

**A=*fair* B=*poor OR* A3 vs Within**



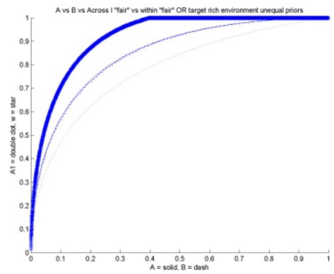

**A=*poor* B=*good OR* A3 vs Within**



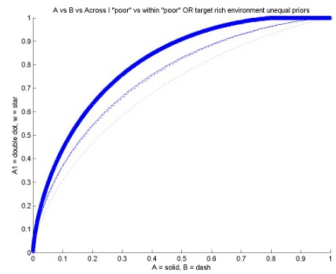

**A=*poor* B=*fair OR* A3 vs Within**



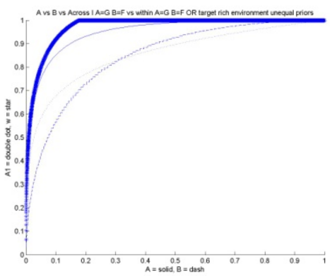

**A=B=*good OR* unequal priors A1 vs Within**



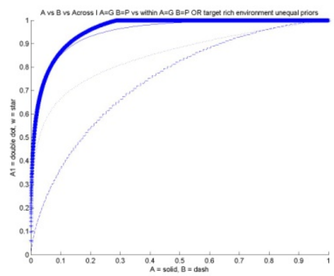

**A=B=*fair OR* unequal priors A1 vs Within**



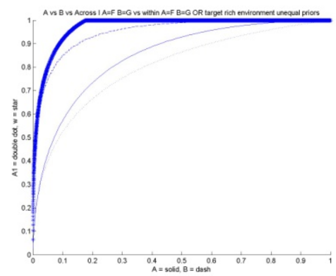

**A=B=*poor OR* unequal priors A1 vs Within**



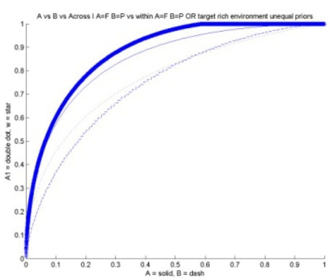

**A=*good* B=*fair OR* unequal priors A1 vs Within**



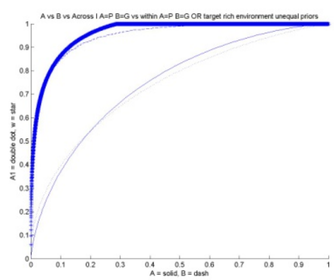

**A=*good* B=*poor OR* unequal priors A1 vs Within**



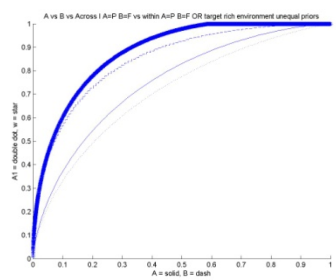

**A=*fair* B=*good OR* unequal priors A1 vs Within**



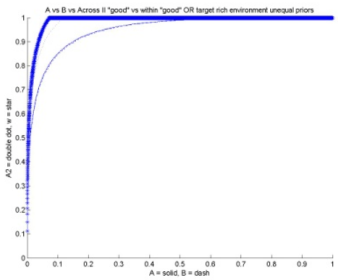

**A=*fair* B=*poor OR* unequal priors A1 vs Within**



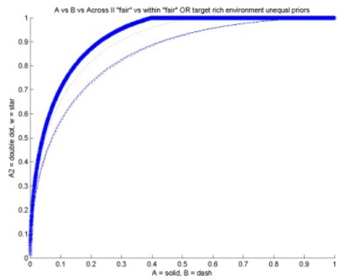

**A=*poor* B=*good OR* unequal priors A1 vs Within**



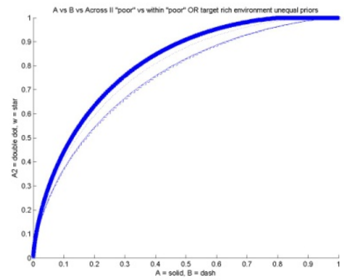

**A=*poor* B=*fair OR* unequal priors A1 vs Within**



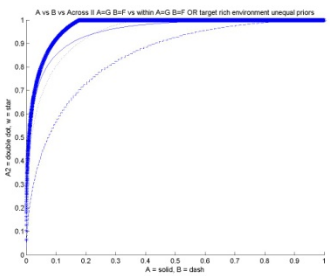

**A=B=*good OR* unequal priors A2 vs Within**



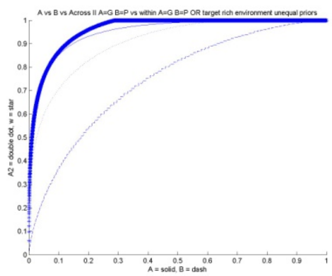

**A=B=*fair OR* unequal priors A2 vs Within**



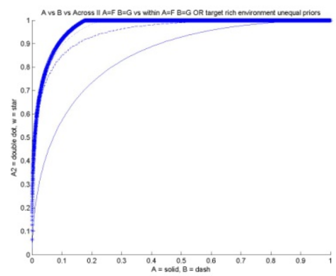

**A=B=*poor OR* unequal priors A2 vs Within**



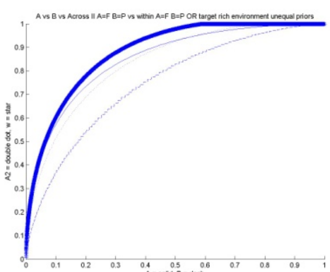

**A=*good* B=*fair OR* unequal priors A2 vs Within**



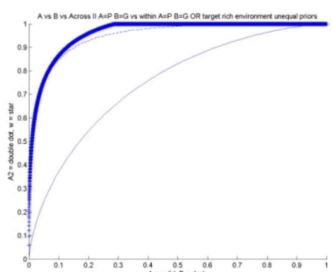

**A=*good* B=*poor OR* unequal priors A2 vs Within**



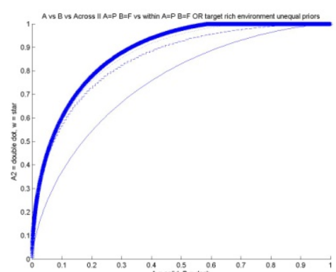

**A=*fair* B=*good OR* unequal priors A2 vs Within**



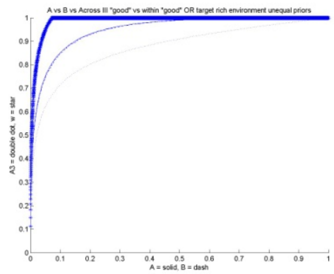

**A=*fair* B=*poor OR* unequal priors A2 vs Within**



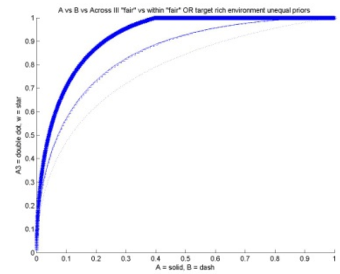

**A=*poor* B=*good OR* unequal priors A2 vs Within**



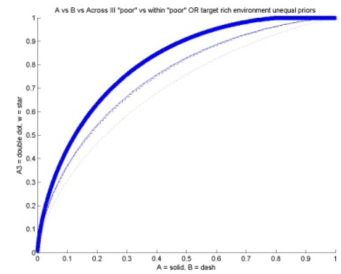

**A=*poor* B=*fair OR* unequal priors A2 vs Within**



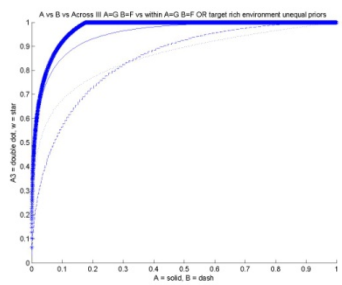

**A=B=*good OR* unequal priors A3 vs Within**



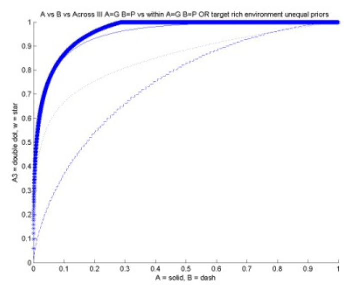

**A=B=*fair OR* unequal priors A3 vs Within**



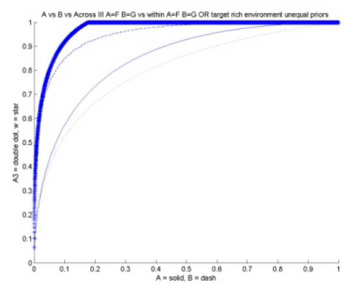

**A=B=*poor OR* unequal priors A3 vs Within**



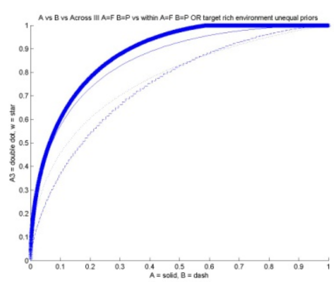

**A=*good* B=*fair OR* unequal priors A3 vs Within**



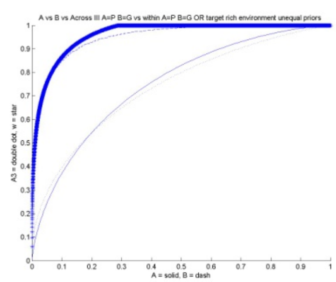

**A=*good* B=*poor OR* unequal priors A3 vs Within**



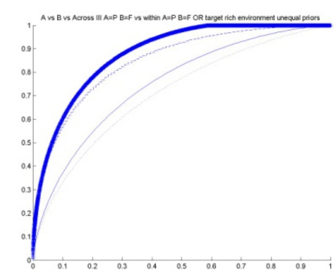

**A=*fair* B=*good OR* unequal priors A3 vs Within**




**A=*fair* B=*poor OR* unequal priors A3 vs Within**

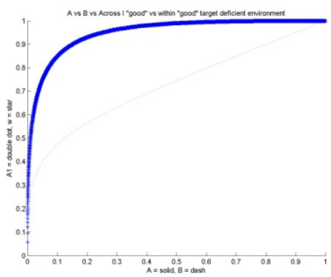



**A=*poor* B=*good OR* unequal priors A3 vs Within**

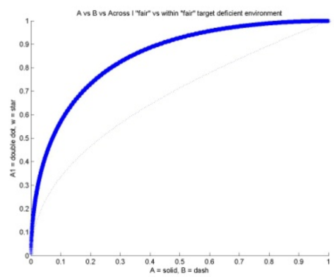



**A=*poor* B=*fair OR* unequal priors A3 vs Within**

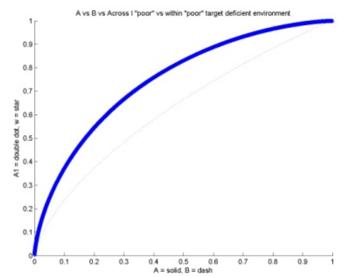
*A.3 – Deficient Target Environment Figures*



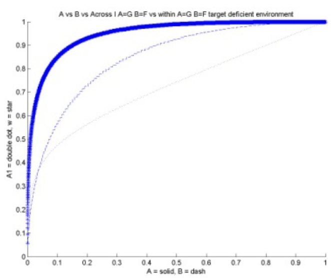

**A=B=*good AND* A1 vs Within**



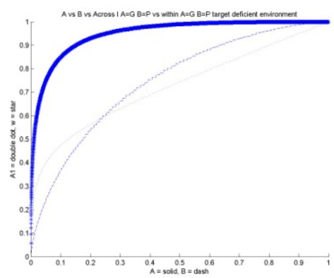

**A=B=*fair AND* A1 vs Within**



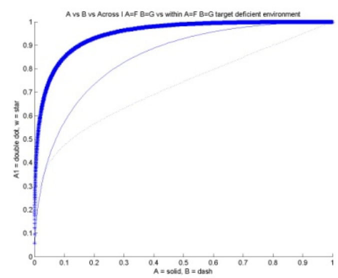

**A=B=*poor AND* A1 vs Within**



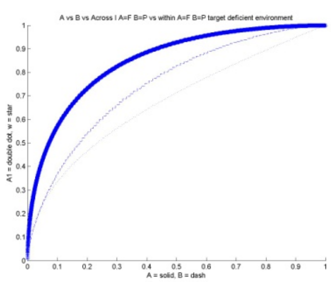

**A=*good* B=*fair AND* A1 vs Within**



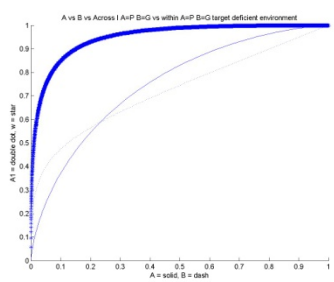

**A=*good* B=*poor AND* A1 vs Within**



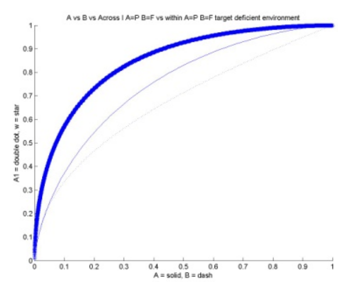

**A=*fair* B=*good AND* A1 vs Within**



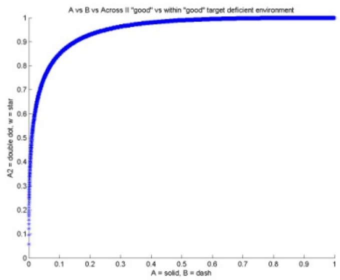

**A=*fair* B=*poor AND* A1 vs Within**



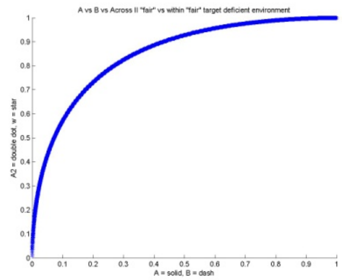

**A=*poor* B=*good AND* A1 vs Within**



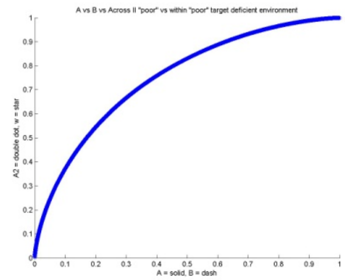

**A=*poor* B=*fair AND* A1 vs Within**



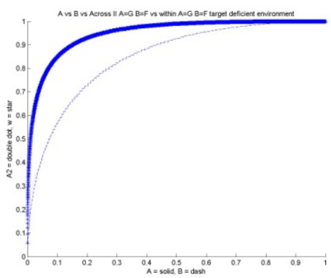

**A=B=*good AND* A2 vs Within**



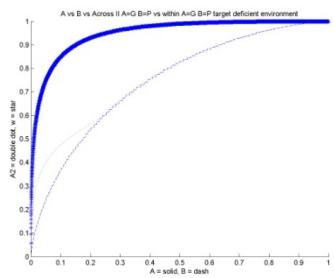

**A=B=*fair AND* A2 vs Within**



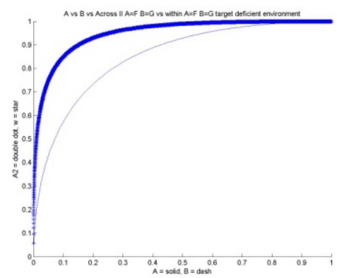

**A=B=*poor AND* A2 vs Within**



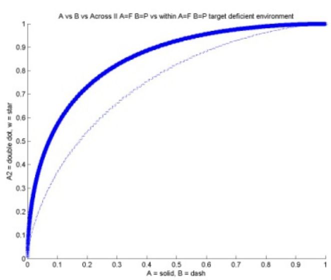
**A=*good* B=*fair AND* A2 vs Within**



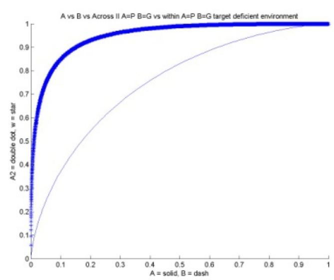
**A=*good* B=*poor AND* A2 vs Within**



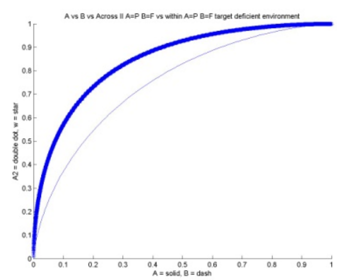
**A=*fair* B=*good AND* A2 vs Within**



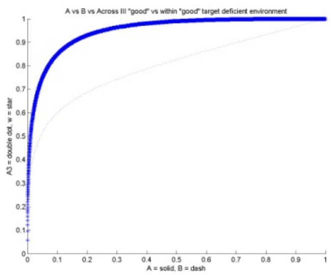
**A=*fair* B=*poor AND* A2 vs Within**



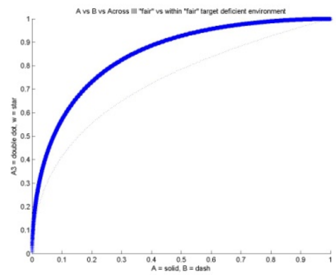
**A=*poor* B=*good AND* A2 vs Within**



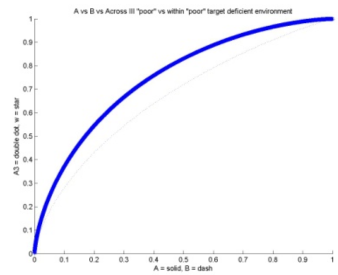
**A=*poor* B=*fair AND* A2 vs Within**



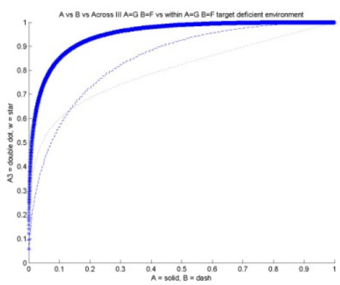

**A=B=*good AND* A3 vs Within**



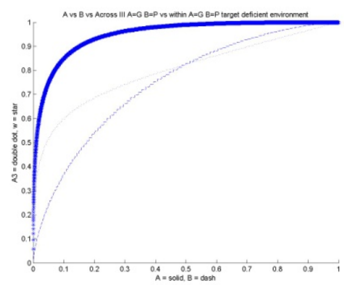

**A=B=*fair AND* A3 vs Within**



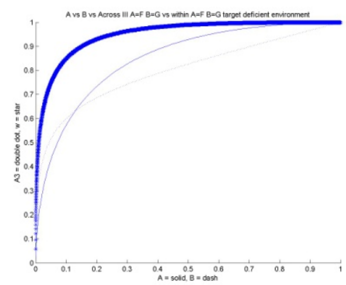

**A=B=*poor AND* A3 vs Within**



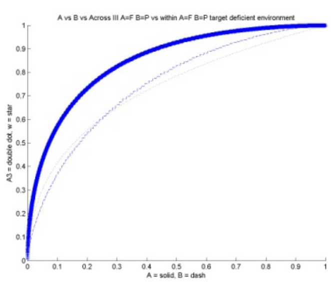

**A=*good* B=*fair AND* A3 vs Within**



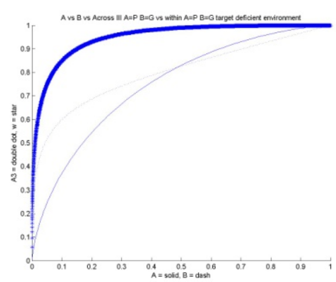

**A=*good* B=*poor AND* A3 vs Within**



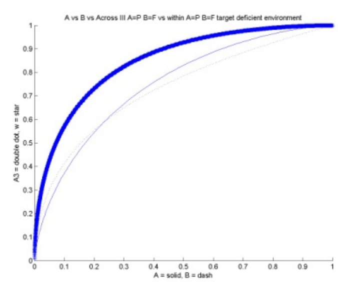

**A=*fair* B=*good AND* A3 vs Within**



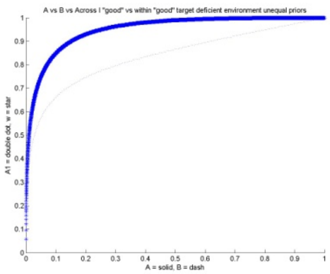

**A=*fair* B=*poor AND* A3 vs Within**

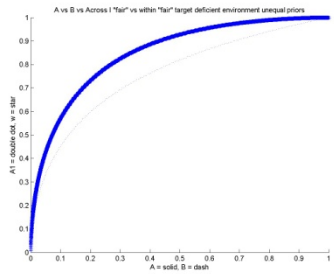


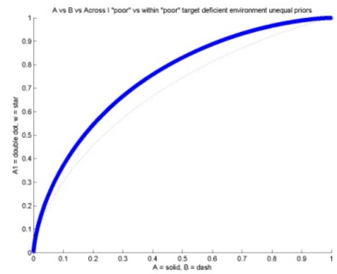

**A=*poor* B=*good AND* A3 vs Within**




**A=*poor* B=*fair AND* A3 vs Within**



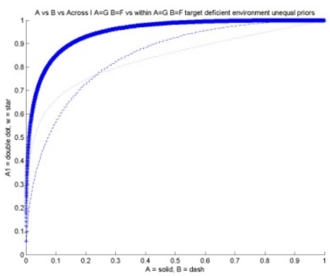



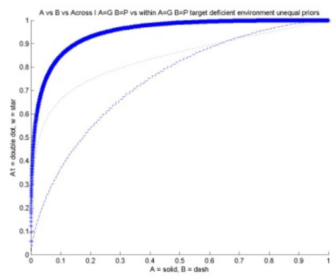



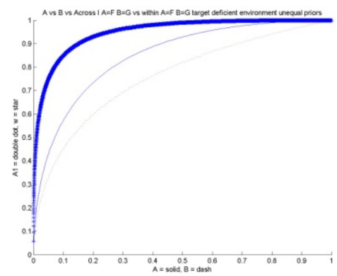

**A=B=*good AND* unequal priors A1 vs Within**

**A=B=*fair AND* unequal priors A1 vs Within**

**A=B=*poor AND* unequal priors A1 vs Within**



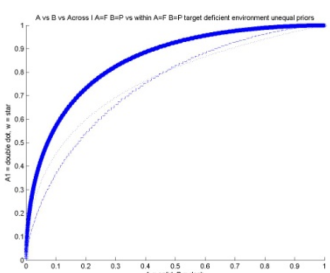



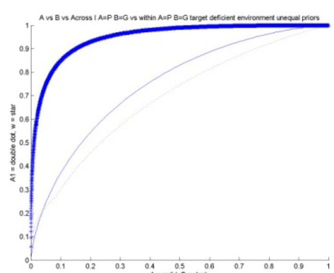



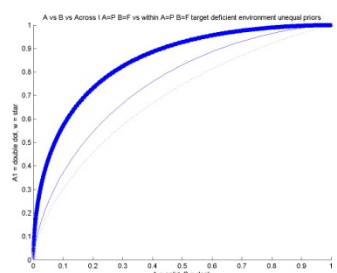

**A=*good* B=*fair AND* unequal priors A1 vs Within**

**A=*good* B=*poor AND* unequal priors A1 vs Within**

**A=*fair* B=*good AND* unequal priors A1 vs Within**



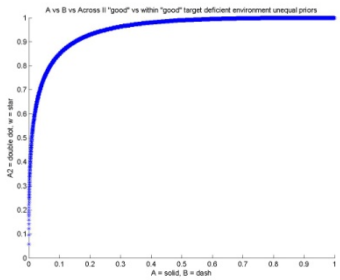



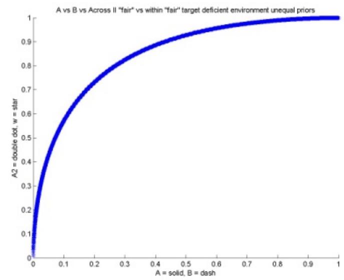

**A=*fair* B=*poor AND* unequal priors A1 vs Within**

**A=*poor* B=*good AND* unequal priors A1 vs Within**

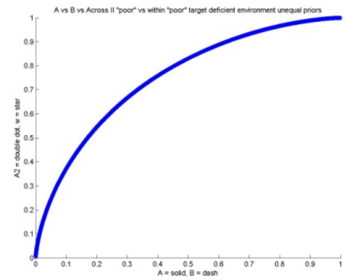
**A=*poor* B=*fair AND* unequal priors A1 vs Within**



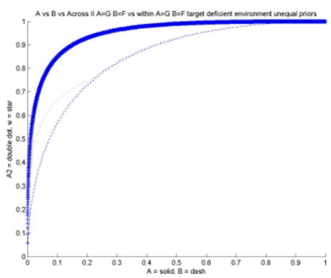

**A=B=*good AND* unequal priors A2 vs Within**



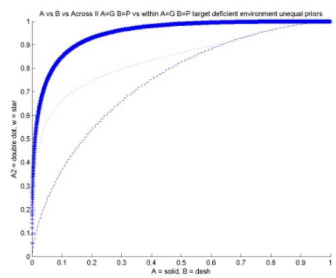

**A=B=*fair AND* unequal priors A2 vs Within**



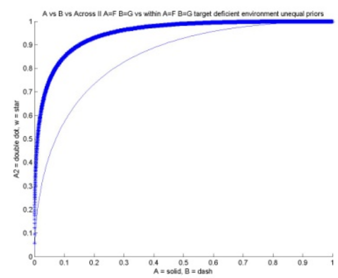

**A=B=*poor AND* unequal priors A2 vs Within**



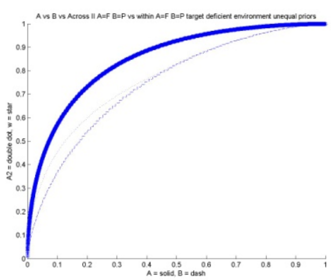

**A=*good* B=*fair AND* unequal priors A2 vs Within**



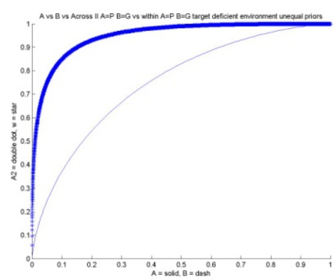

**A=*good* B=*poor AND* unequal priors A2 vs Within**



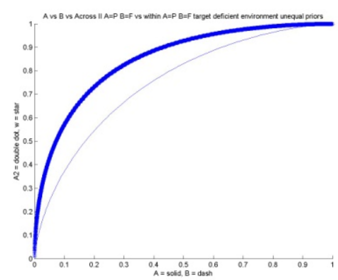

**A=*fair* B=*good AND* unequal priors A2 vs Within**



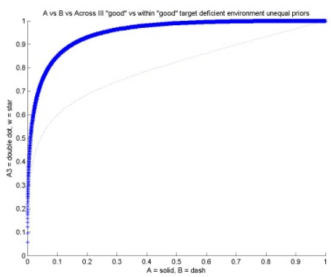

**A=*fair* B=*poor AND* unequal priors A2 vs Within**



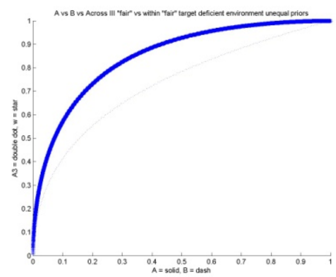

**A=*poor* B=*good AND* unequal priors A2 vs Within**



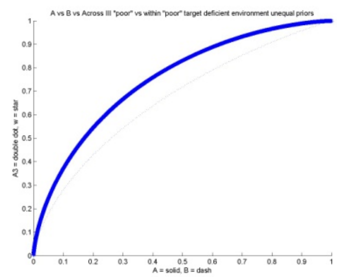

**A=*poor* B=*fair AND* unequal priors A2 vs Within**



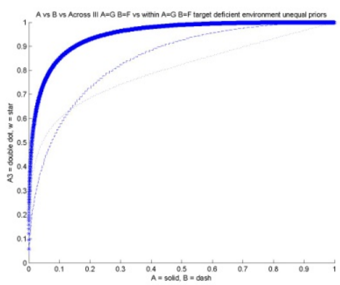

**A=B=*good AND* unequal priors A3 vs Within**



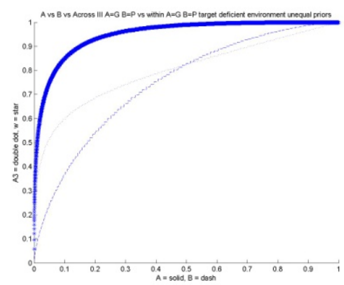

**A=B=*fair AND* unequal priors A3 vs Within**



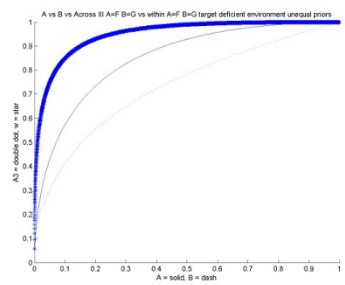

**A=B=*poor AND* unequal priors A3 vs Within**



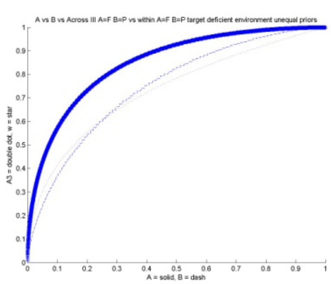

**A=*good* B=*fair AND* unequal priors A3 vs Within**



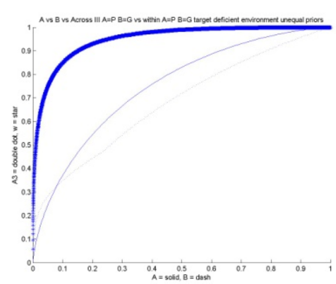

**A=*good* B=*poor AND* unequal priors A3 vs Within**



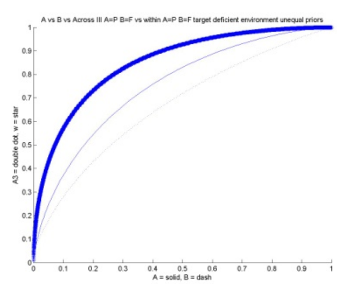

**A=*fair* B=*good AND* unequal priors A3 vs Within**



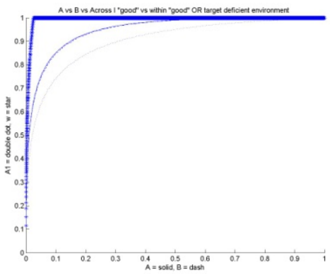

**A=*fair* B=*poor AND* unequal priors A3 vs Within**



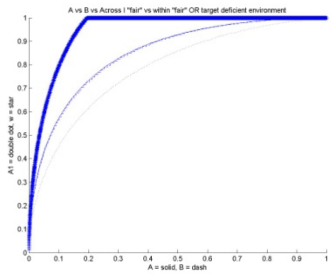

**A=*poor* B=*good AND* unequal priors A3 vs Within**



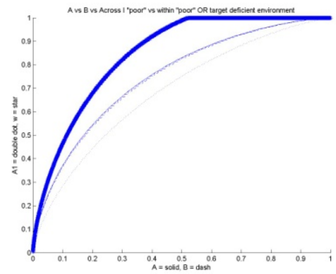

**A=*poor* B=*fair AND* unequal priors A3 vs Within**



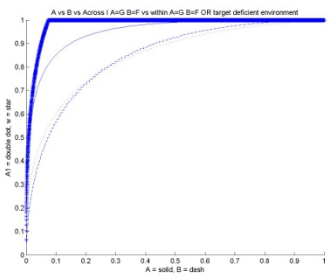

**A=B=*good OR* A1 vs Within**



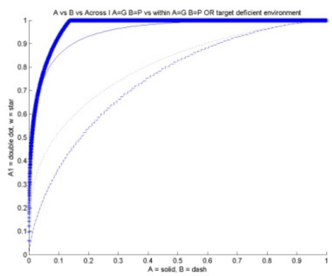

**A=B=*fair OR* A1 vs Within**



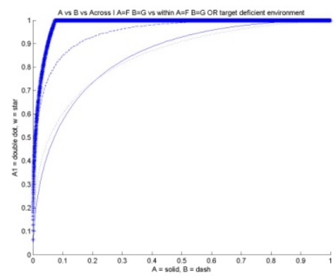

**A=B=*poor OR* A1 vs Within**



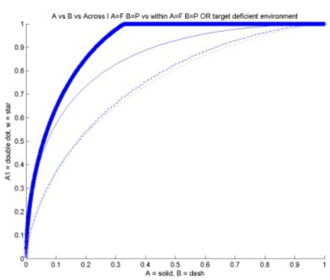

**A=*good* B=*fair OR* A1 vs Within**



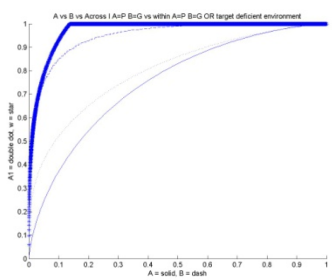

**A=*good* B=*poor OR* A1 vs Within**



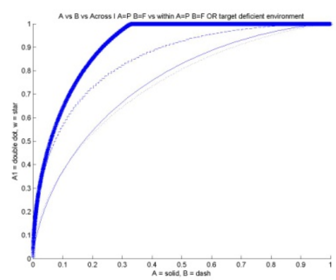

**A=*fair* B=*good OR* A1 vs Within**



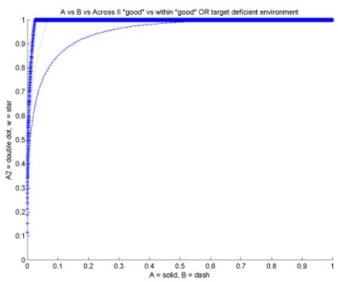

**A=*fair* B=*poor OR* A1 vs Within**



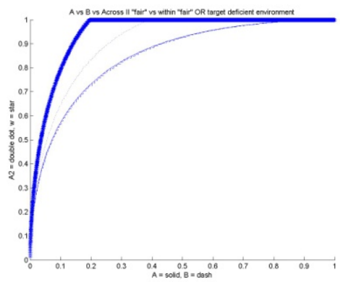

**A=*poor* B=*good OR* A1 vs Within**



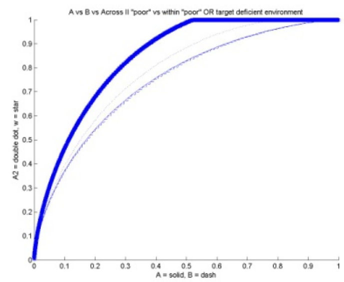

**A=*poor* B=*fair OR* A1 vs Within**



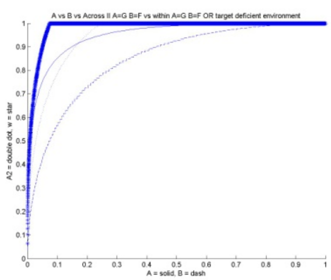

**A=B=*good OR* A2 vs Within**



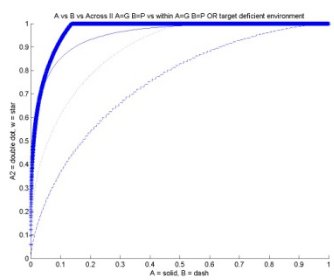

**A=B=*fair OR* A2 vs Within**



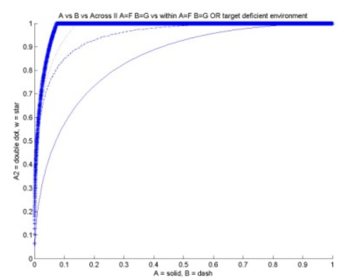

**A=B=*poor OR* A2 vs Within**



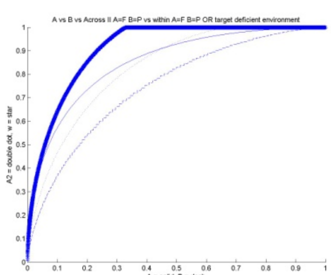

**A=*good* B=*fair OR* A2 vs Within**



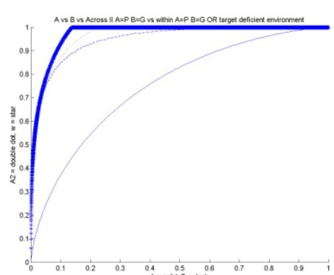

**A=*good* B=*poor OR* A2 vs Within**



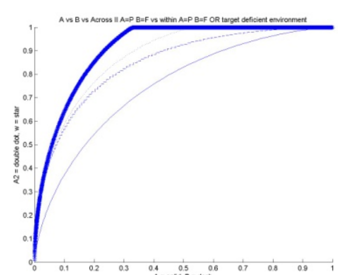

**A=*fair* B=*good OR* A2 vs Within**



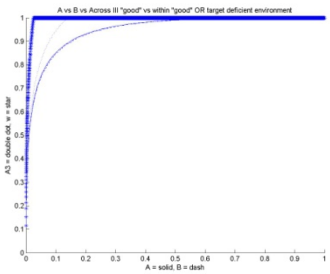

**A=*fair* B=*poor OR* A2 vs Within**



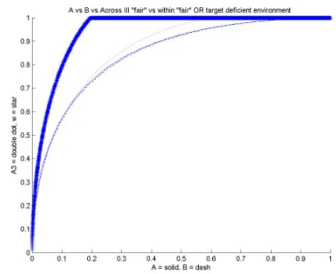

**A=*poor* B=*good OR* A2 vs Within**



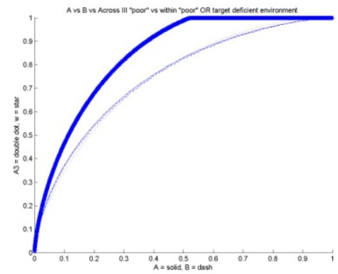

**A=*poor* B=*fair OR* A2 vs Within**



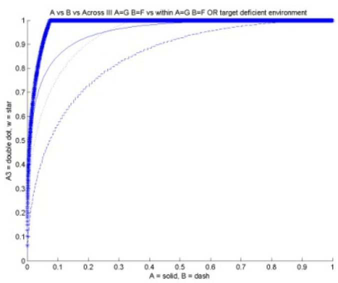

**A=B=*good OR* A3 vs Within**



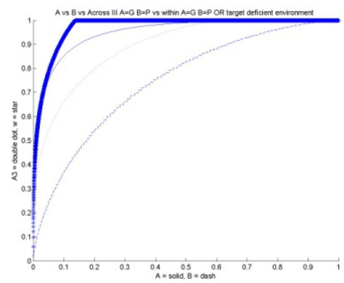

**A=B=*fair OR* A3 vs Within**



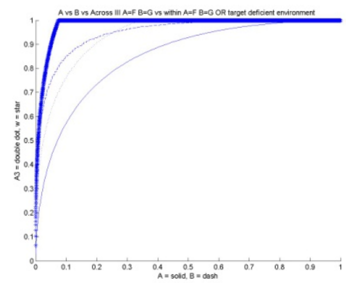

**A=B=*poor OR* A3 vs Within**



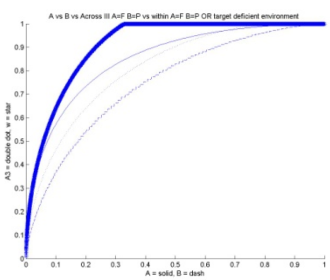

**A=*good* B=*fair OR* A3 vs Within**



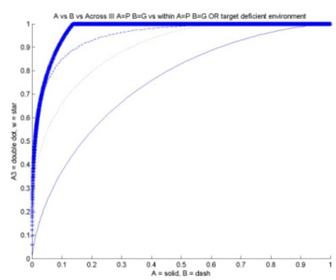

**A=*good* B=*poor OR* A3 vs Within**



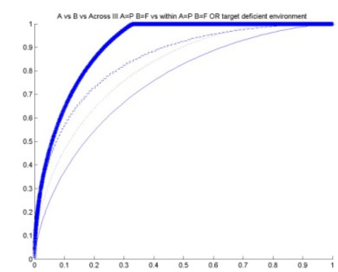

**A=*fair* B=*good OR* A3 vs Within**



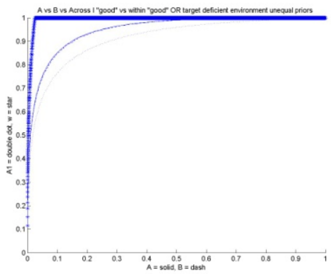

**A=*fair* B=*poor OR* A3 vs Within**



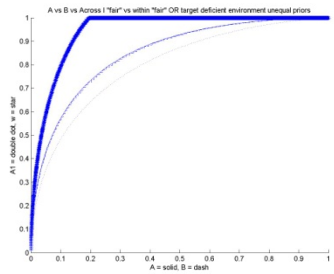

**A=*poor* B=*good OR* A3 vs Within**



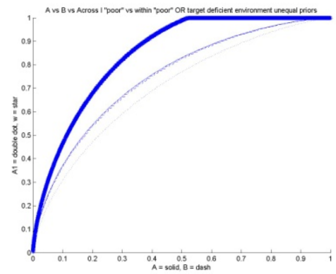

**A=*poor* B=*fair OR* A3 vs Within**



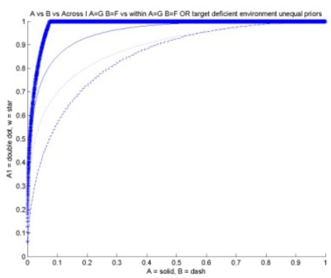

**A=B=*good OR* unequal priors A1 vs Within**



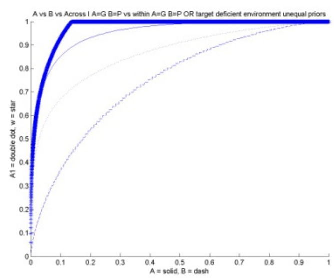

**A=B=*fair OR* unequal priors A1 vs Within**



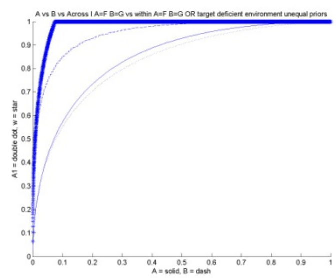

**A=B=*poor OR* unequal priors A1 vs Within**



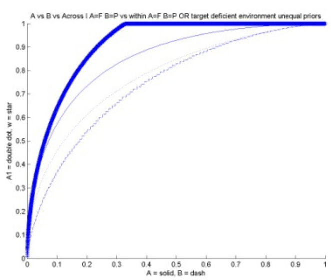

**A=*good* B=*fair OR* unequal priors A1 vs Within**



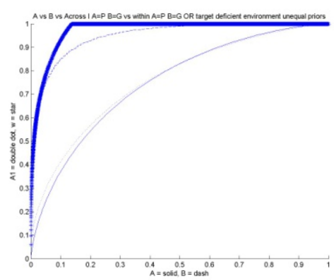

**A=*good* B=*poor OR* unequal priors A1 vs Within**



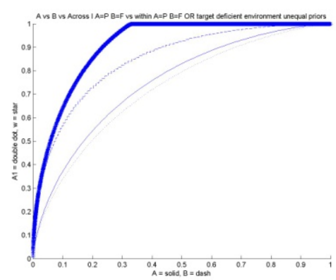

**A=*fair* B=*good OR* unequal priors A1 vs Within**



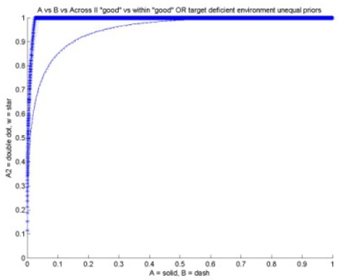

**A=*fair* B=*poor OR* unequal priors A1 vs Within**



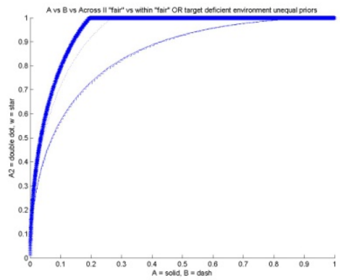

**A=*poor* B=*good OR* unequal priors A1 vs Within**



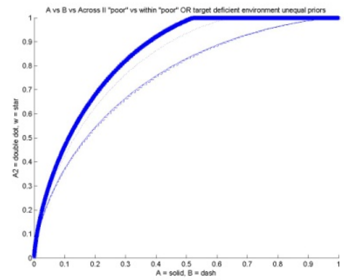

**A=*poor* B=*fair OR* unequal priors A1 vs Within**



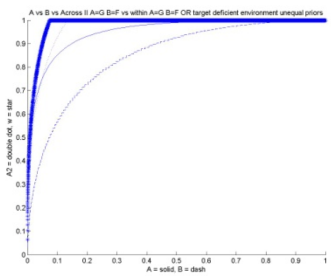
**A=B=*good OR* unequal priors A2 vs Within**



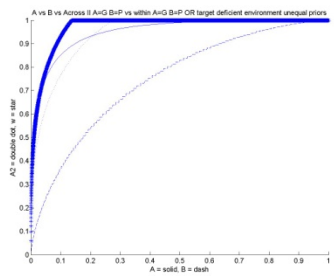
**A=B=*fair OR* unequal priors A2 vs Within**



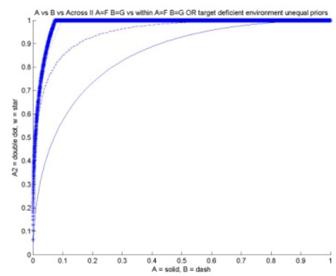
**A=B=*poor OR* unequal priors A2 vs Within**



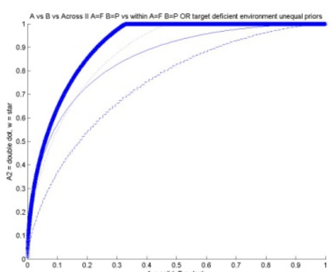
**A=*good* B=*fair OR* unequal priors A2 vs Within**



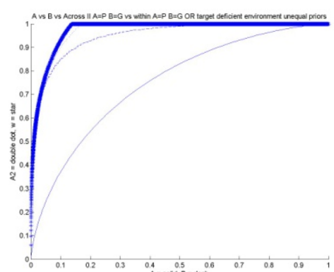
**A=*good* B=*poor OR* unequal priors A2 vs Within**



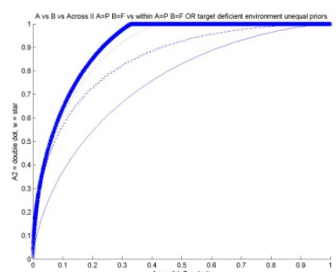
**A=*fair* B=*good OR* unequal priors A2 vs Within**



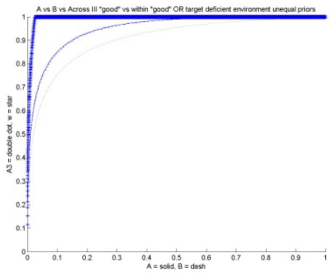
**A=*fair* B=*poor OR* unequal priors A2 vs Within**



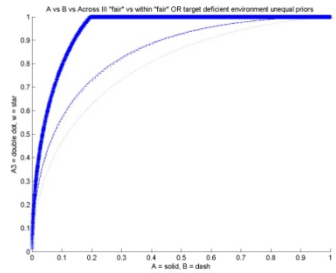
**A=*poor* B=*good OR* unequal priors A2 vs Within**



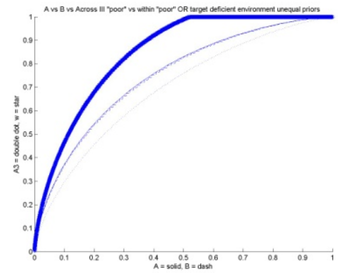
**A=*poor* B=*fair OR* unequal priors A2 vs Within**



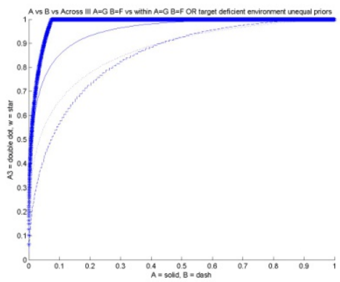

**A=B=*good OR* unequal priors A3 vs Within**



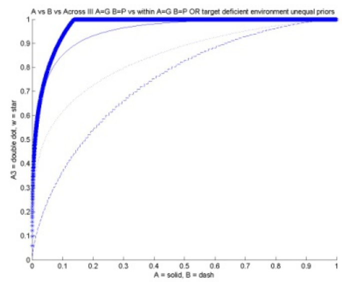

**A=B=*fair OR* unequal priors A3 vs Within**



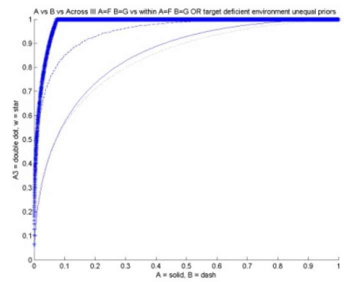

**A=B=*poor OR* unequal priors A3 vs Within**



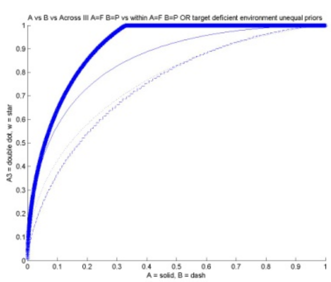

**A=*good* B=*fair OR* unequal priors A3 vs Within**



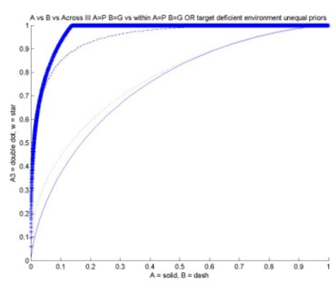

**A=*good* B=*poor OR* unequal priors A3 vs Within**



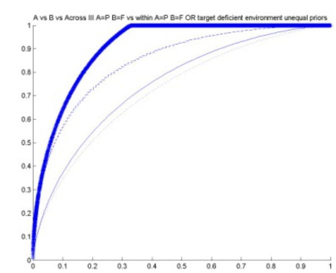

**A=*fair* B=*good OR* unequal priors A3 vs Within**




**A=*fair* B=*poor OR* unequal priors A3 vs Within**




**A=*poor* B=*good OR* unequal priors A3 vs Within**




**A=*poor* B=*fair OR* unequal priors A3 vs Within**

## *Appendix B*

Included in this appendix are all of the bias curves generated during the simulation. As a quick outline, in each figure four bias curves are plotted: the uncorrelated curve and the three positively or negatively correlated curves which are specified by the $\rho \geq 0$ or $\rho \leq 0$ in the caption below. The Boolean rule along with the distribution of prior target probability is also included in the captions. The figures will be presented in the following order:

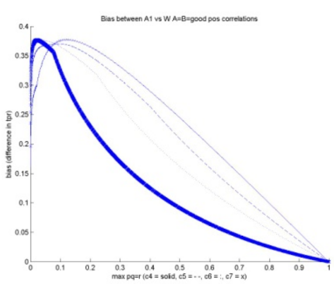
1) *Enhanced Target Prevalence*
    a) *Across I* versus *Within*
    b) *Across II* versus *Within*
    c) *Across III* versus *Within*
2) *Rich Target Prevalence*
    a) *Across I* versus *Within*
    b) *Across II* versus *Within*
    c) *Across III* versus *Within*
3) *Deficient Target Prevalence*
    a) *Across I* versus *Within*
    b) *Across II* versus *Within*
    c) *Across III* versus *Within*

As each figure is composed of four different bias curves (one for each level of positive or negative correlation), consider the following table for reference purposes.

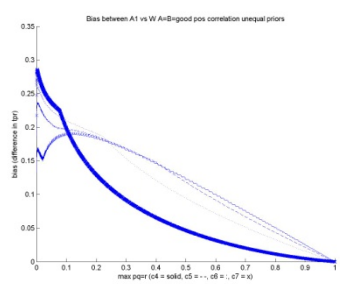
| Correlation Coefficient | Plot Symbol |
|---|---|
| ρ = -0.8 | x (bold curve) |
| ρ = -0.5 | : (double dot) |
| ρ = -0.3 | -- (double dash) |
| ρ = 0 | smooth line |
| ρ = 0.3 | -- (double dash) |
| ρ = 0.5 | : (double dot) |
| ρ = 0.8 | x (bold curve) |

*B.1 – Enhanced Target Prevalence*

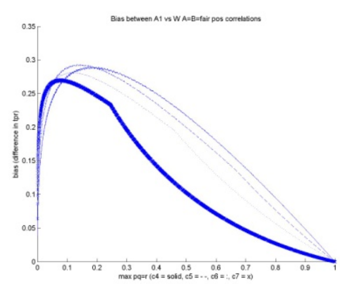
*B.1.1 – Across I* versus *Within*



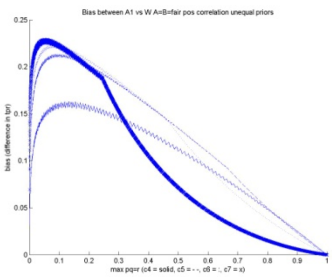

**A=B=good *AND* ρ ≥ 0**



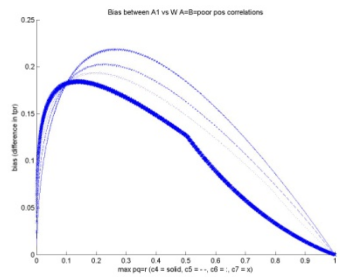

**A=B=good *AND* unequal priors ρ ≥ 0**



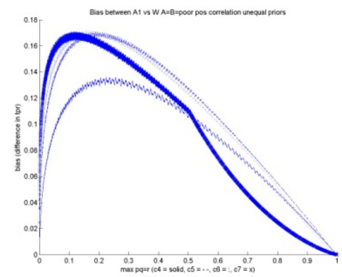

**A=B=fair *AND* ρ ≥ 0**



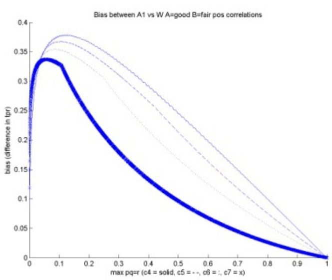

**A=B=fair *AND* unequal priors ρ ≥ 0**



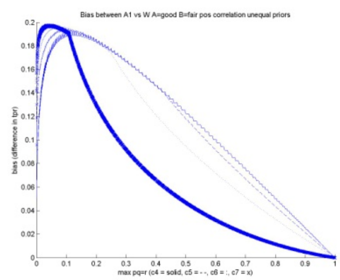

**A=B=poor *AND* ρ ≥ 0**



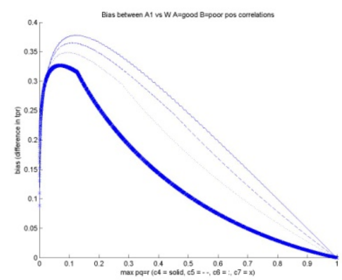

**A=B=poor unequal priors *AND* ρ ≥ 0**



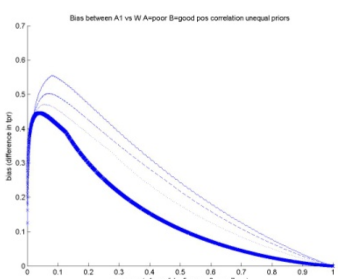

**A=good B=fair *AND* ρ ≥ 0**



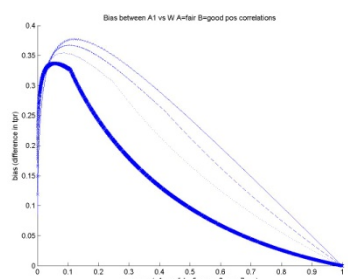

**A=good B=fair *AND* unequal priors ρ ≥ 0**



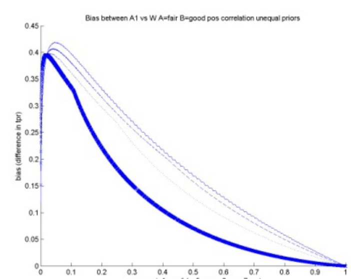

**A=good B=poor *AND* ρ ≥ 0**



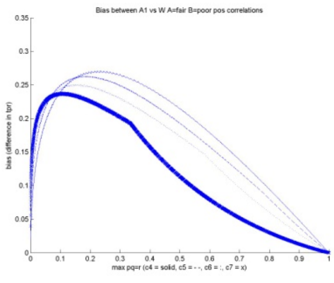

**A=good B=poor *AND* unequal priors ρ ≥ 0**



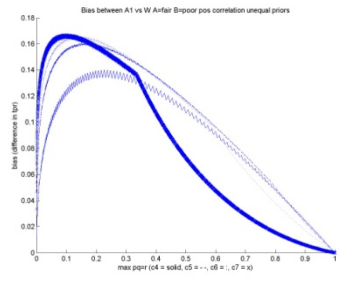

**A=fair B=good *AND* ρ ≥ 0**



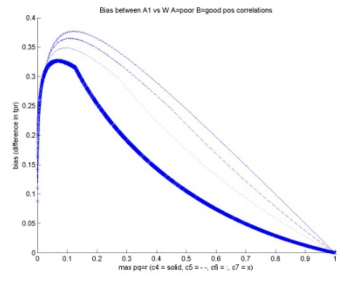

**A=fair B=good *AND* unequal priors ρ ≥ 0**



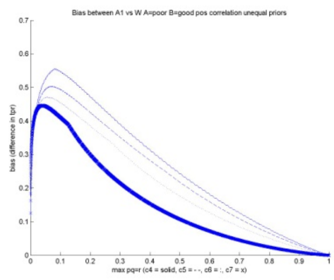

**A=fair B=poor *AND* ρ ≥ 0**



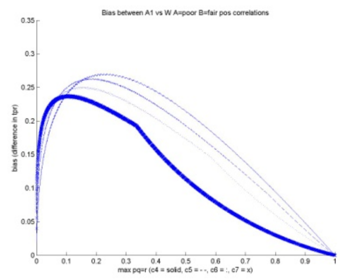

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



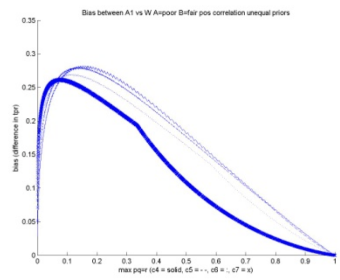

**A=poor B=good *AND* ρ ≥ 0**



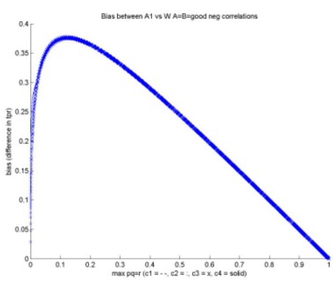

**A=poor B=good *AND* unequal priors ρ ≥ 0**



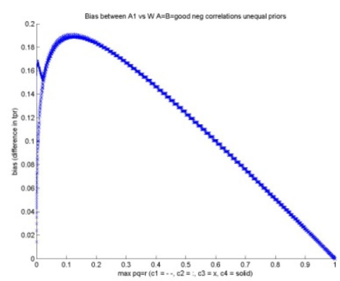

**A=poor B=fair *AND* ρ ≥ 0**



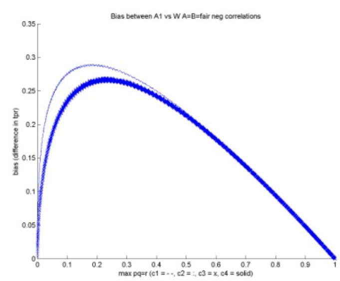

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



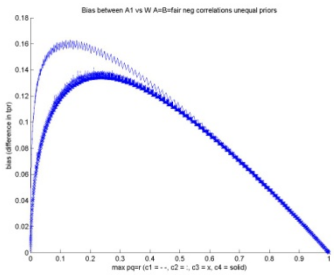

**A=B=good *AND* ρ ≤ 0**



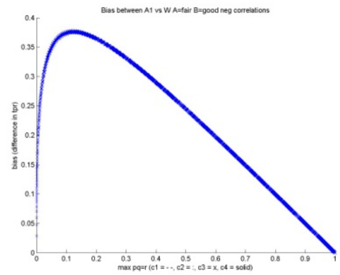

**A=B=good *AND* unequal priors ρ ≤ 0**



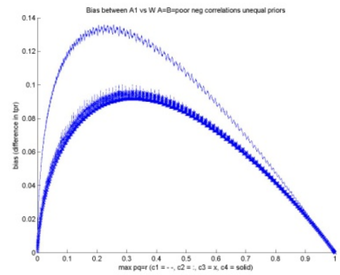

**A=B=fair *AND* ρ ≤ 0**



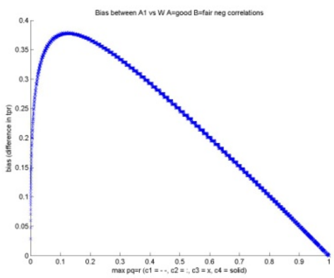
**A=B=fair *AND* unequal priors ρ ≤ 0**



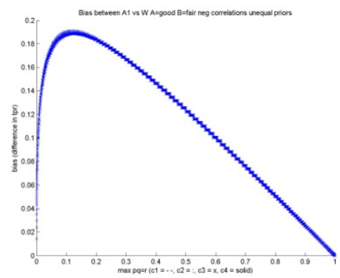
**A=B=poor *AND* ρ ≤ 0**



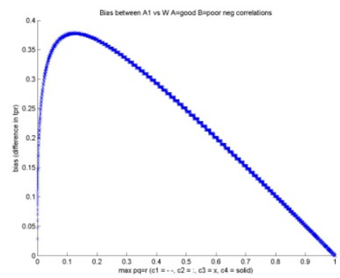
**A=B=poor unequal priors *AND* ρ ≤ 0**



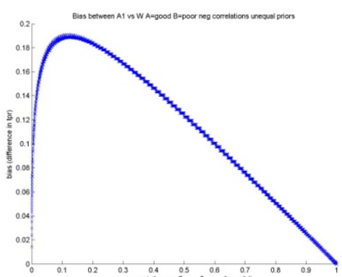
**A=good B=fair *AND* ρ ≤ 0**



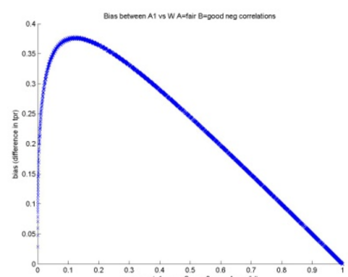
**A=good B=fair *AND* unequal priors ρ ≤ 0**



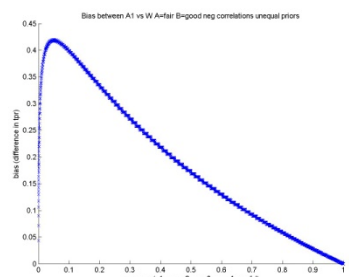
**A=good B=poor *AND* ρ ≤ 0**



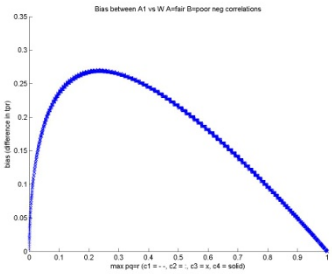
**A=good B=poor *AND* unequal priors ρ ≤ 0**



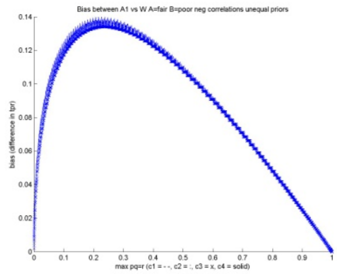
**A=fair B=good *AND* ρ ≤ 0**



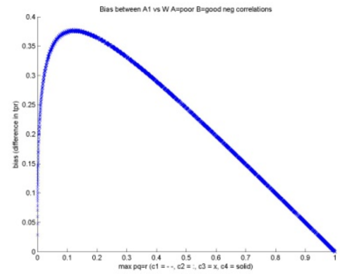
**A=fair B=good *AND* unequal priors ρ ≤ 0**



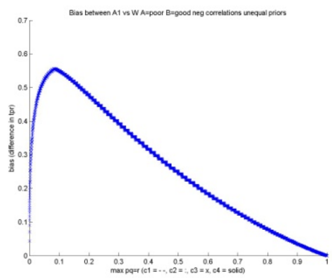

**A=fair B=poor *AND* ρ ≤ 0**



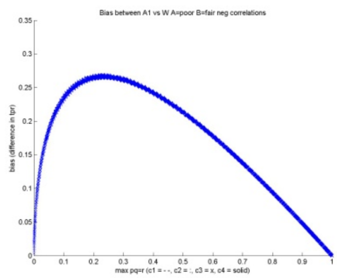

**A=fair B=poor *AND* unequal priors ρ ≤ 0**



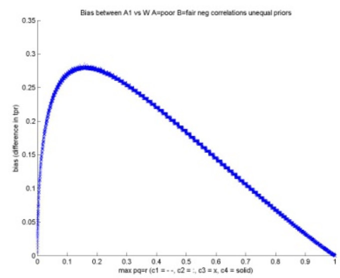

**A=poor B=good *AND* ρ ≤ 0**



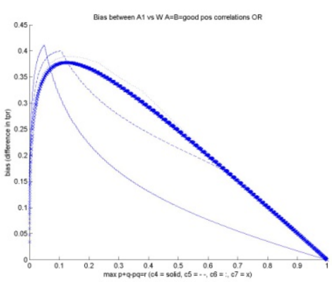

**A=poor B=good *AND* unequal priors ρ ≤ 0**



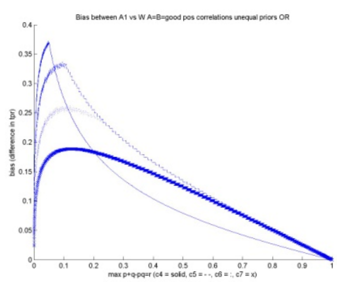

**A=poor B=fair *AND* ρ ≤ 0**



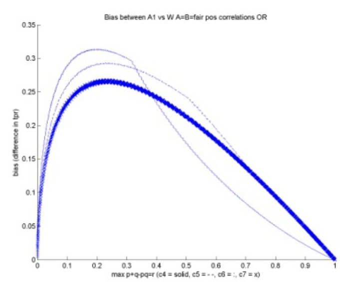

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



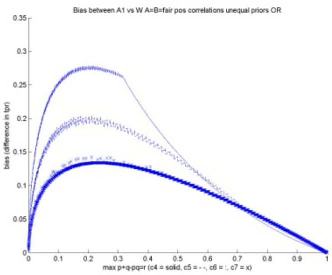

**A=B=good *OR* ρ ≥ 0**



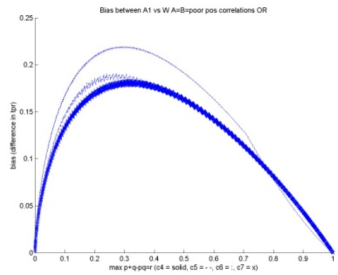

**A=B=good *OR* unequal priors ρ ≥ 0**



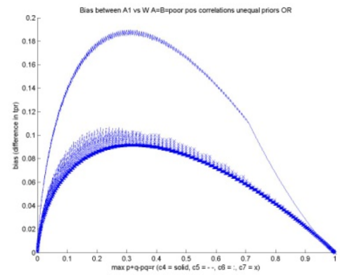

**A=B=fair *OR* ρ ≥ 0**



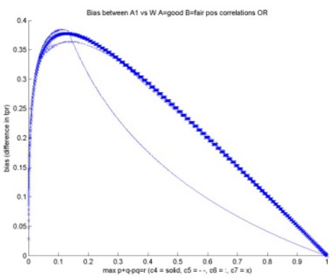

**A=B=fair *OR* unequal priors ρ ≥ 0**



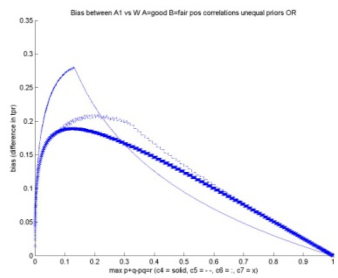

**A=B=poor *OR* ρ ≥ 0**



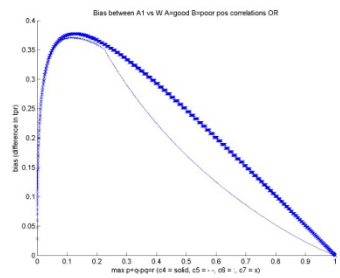

**A=B=poor unequal priors *OR* ρ ≥ 0**



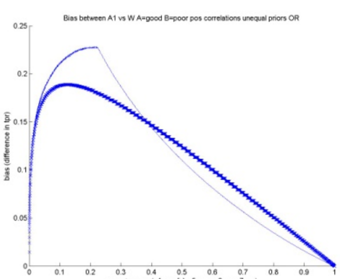

**A=good B=fair *OR* ρ ≥ 0**



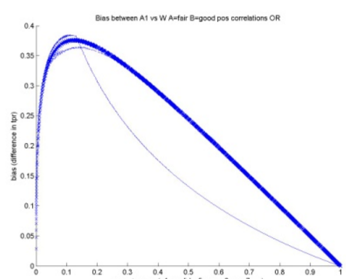

**A=good B=fair *OR* unequal priors ρ ≥ 0**



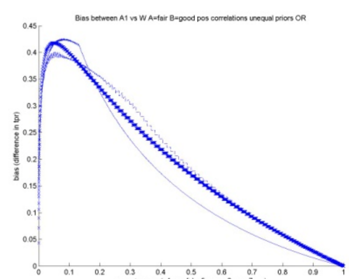

**A=good B=poor *OR* ρ ≥ 0**



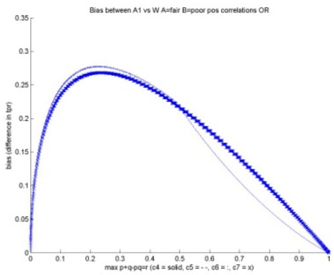

**A=good B=poor *OR* unequal priors ρ ≥ 0**



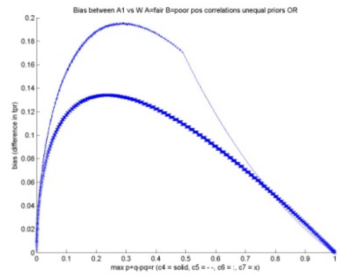

**A=fair B=good *OR* ρ ≥ 0**



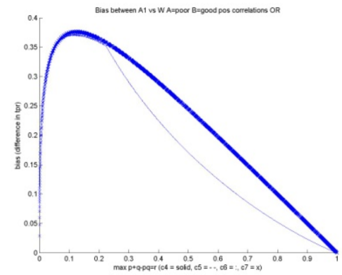

**A=fair B=good *OR* unequal priors ρ ≥ 0**



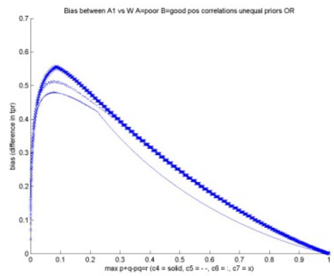

**A=fair B=poor *OR* ρ ≥ 0**



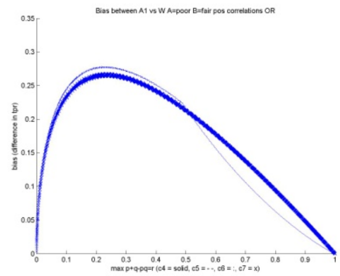

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



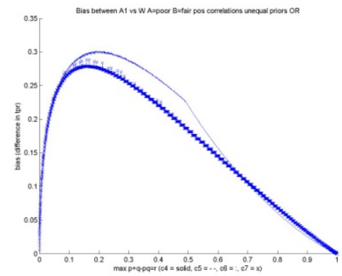

**A=poor B=good *OR* ρ ≥ 0**



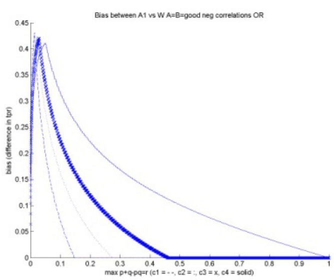

**A=poor B=good *OR* unequal priors ρ ≥ 0**



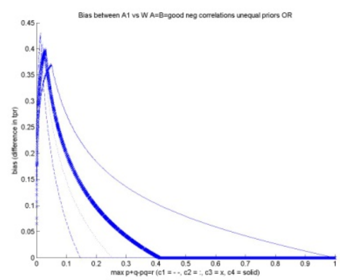

**A=poor B=fair *OR* ρ ≥ 0**



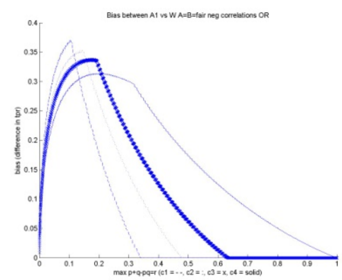

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



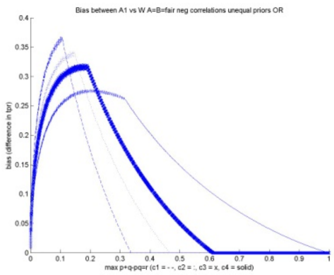

**A=B=good *OR* ρ ≤ 0**



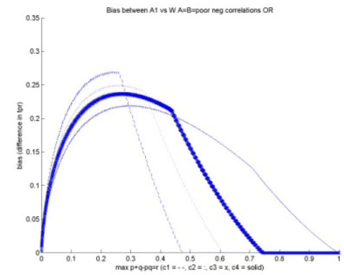

**A=B=good *OR* unequal priors ρ ≤ 0**



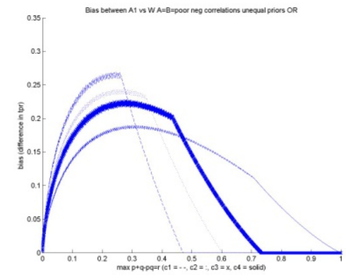

**A=B=fair *OR* ρ ≤ 0**



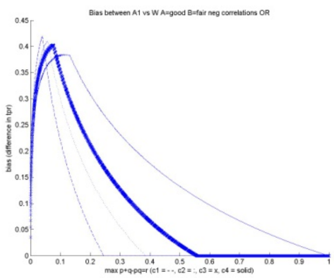

**A=B=fair *OR* unequal priors ρ ≤ 0**



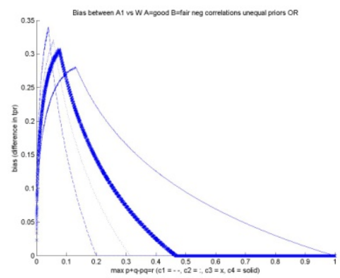

**A=B=poor *OR* ρ ≤ 0**



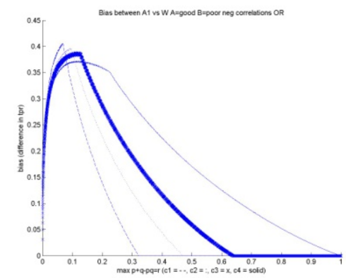

**A=B=poor unequal priors *OR* ρ ≤ 0**



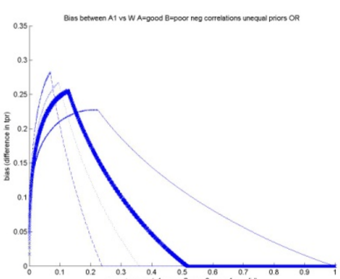

**A=good B=fair *OR* ρ ≤ 0**



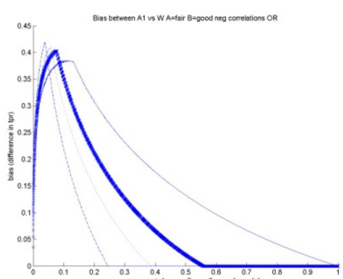

**A=good B=fair *OR* unequal priors ρ ≤ 0**



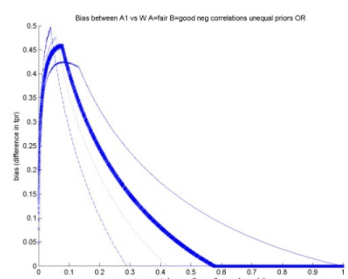

**A=good B=poor *OR* ρ ≤ 0**



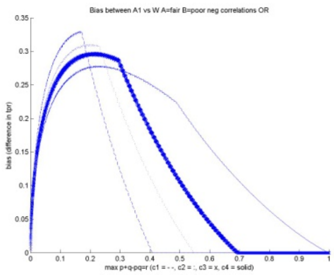

**A=good B=poor *OR* unequal priors ρ ≤ 0**



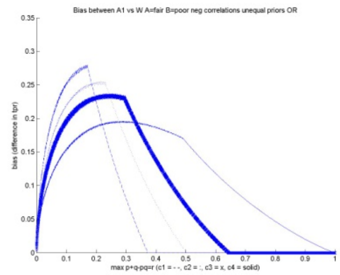

**A=fair B=good *OR* ρ ≤ 0**



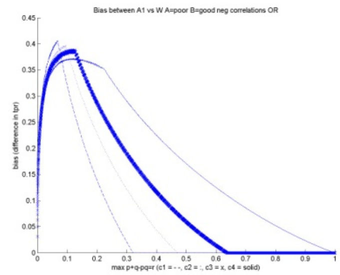

**A=fair B=good *OR* unequal priors ρ ≤ 0**



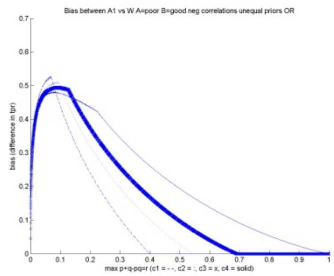

**A=fair B=poor *OR* ρ ≤ 0**



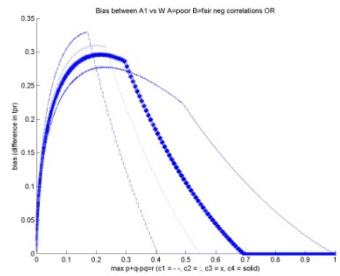

**A=fair B=poor *OR* unequal priors ρ ≤ 0**



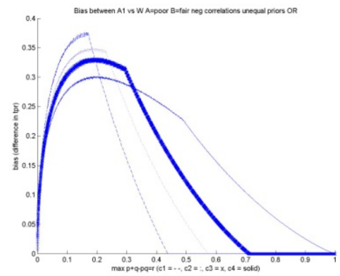

**A=poor B=good *OR* ρ ≤ 0**




**A=poor B=good *OR* unequal priors ρ ≤ 0**

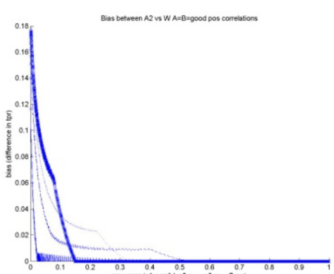



**A=poor B=fair *OR* ρ ≤ 0**

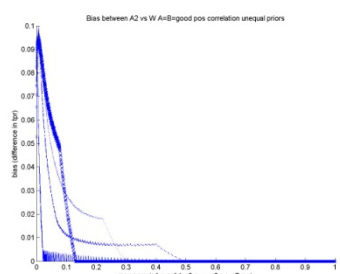



**A=poor B=fair *OR* unequal priors ρ ≤ 0**

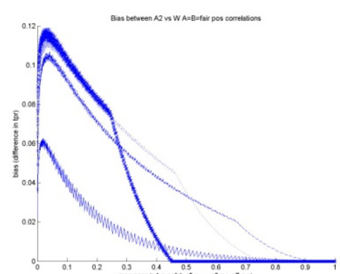
## *B.1.2 – Across II versus Within*



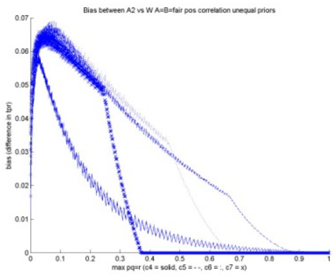

**A=B=good *AND* ρ ≥ 0**



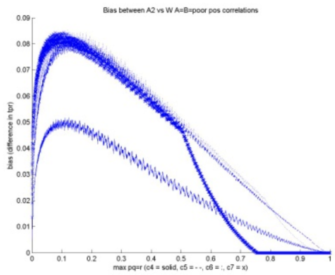

**A=B=good *AND* unequal priors ρ ≥ 0**



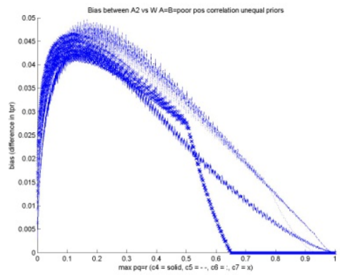

**A=B=fair *AND* ρ ≥ 0**



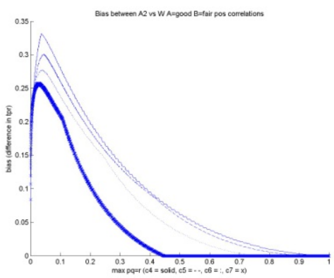

**A=B=fair *AND* unequal priors ρ ≥ 0**



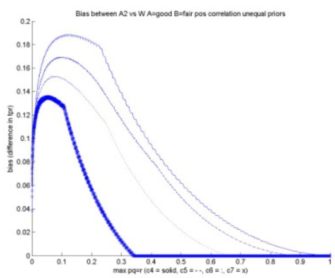

**A=B=poor *AND* ρ ≥ 0**



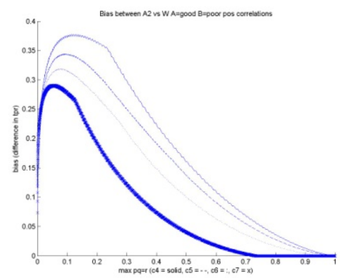

**A=B=poor unequal priors *AND* ρ ≥ 0**



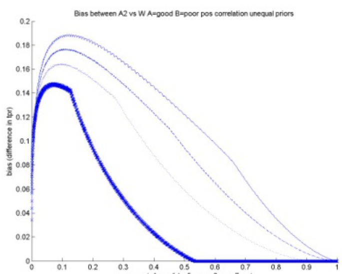

**A=good B=fair *AND* ρ ≥ 0**



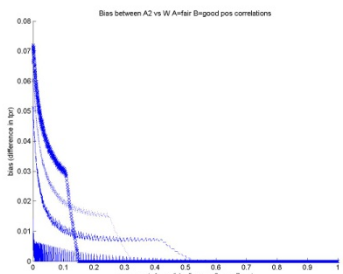

**A=good B=fair *AND* unequal priors ρ ≥ 0**



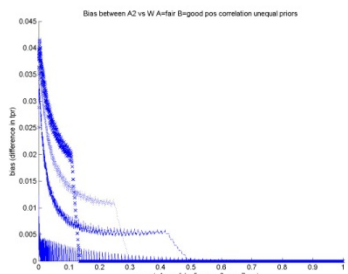

**A=good B=poor *AND* ρ ≥ 0**



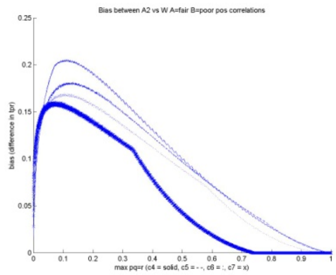

**A=good B=poor *AND* unequal priors ρ ≥ 0**



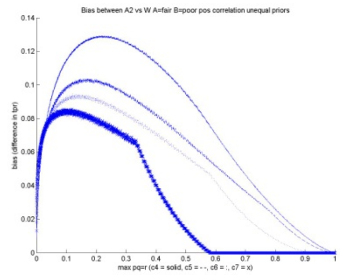

**A=fair B=good *AND* ρ ≥ 0**



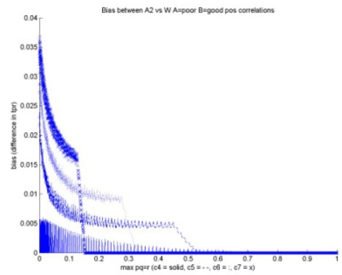

**A=fair B=good *AND* unequal priors ρ ≥ 0**



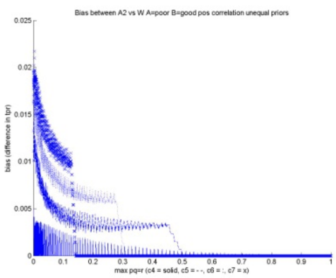

**A=fair B=poor *AND* ρ ≥ 0**



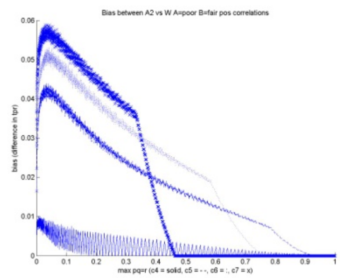

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



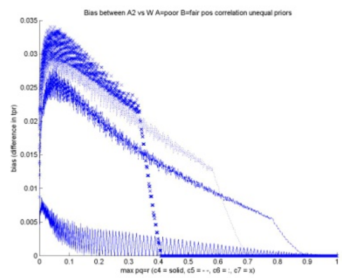

**A=poor B=good *AND* ρ ≥ 0**



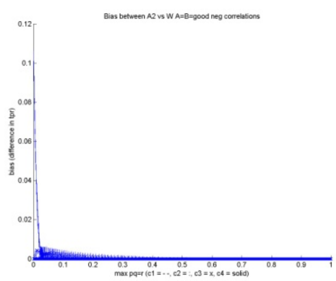

**A=poor B=good *AND* unequal priors ρ ≥ 0**



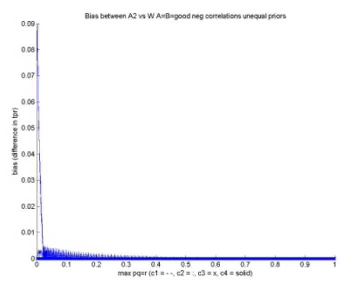

**A=poor B=fair *AND* ρ ≥ 0**



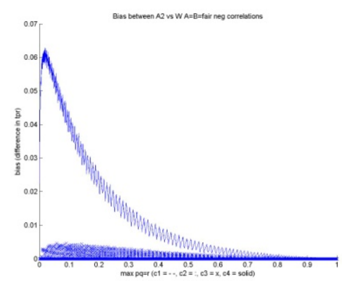

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



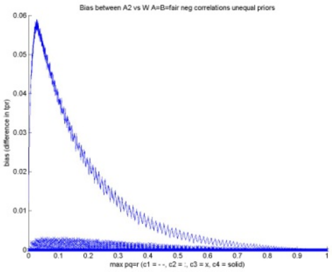

**A=B=good *AND* ρ ≤ 0**



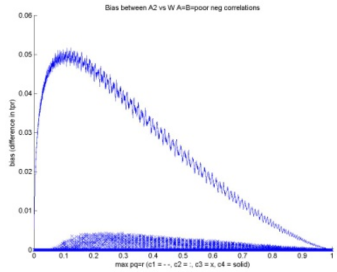

**A=B=good *AND* unequal priors ρ ≤ 0**



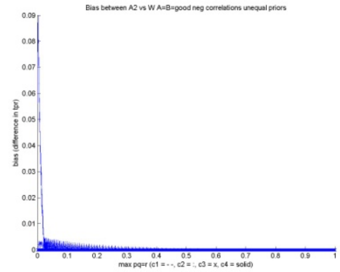

**A=B=fair *AND* ρ ≤ 0**



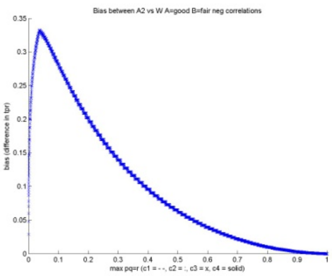

**A=B=fair *AND* unequal priors ρ ≤ 0**



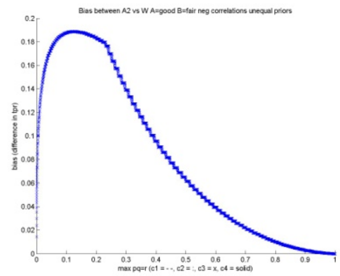

**A=B=poor *AND* ρ ≤ 0**



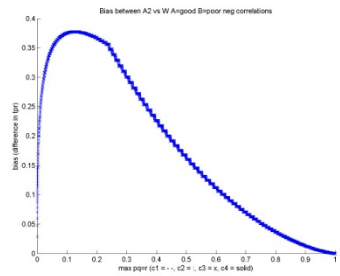

**A=B=poor unequal priors *AND* ρ ≤ 0**



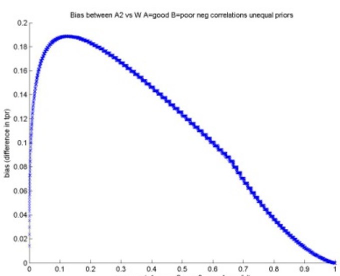

**A=good B=fair *AND* ρ ≤ 0**



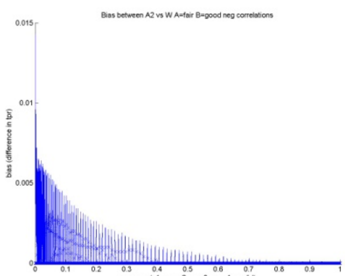

**A=good B=fair *AND* unequal priors ρ ≤ 0**



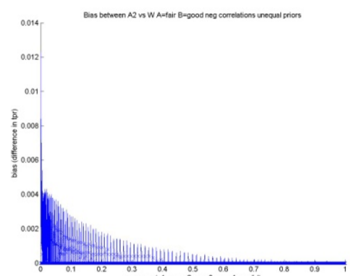

**A=good B=poor *AND* ρ ≤ 0**



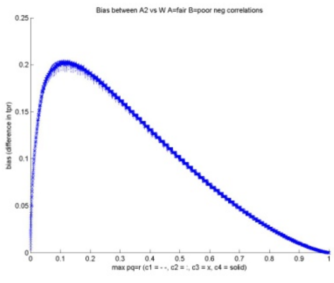

**A=good B=poor *AND* unequal priors ρ ≤ 0**



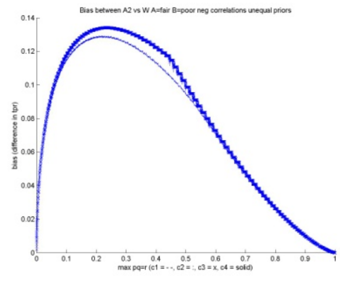

**A=fair B=good *AND* ρ ≤ 0**



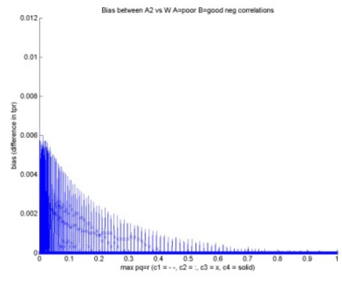

**A=fair B=good *AND* unequal priors ρ ≤ 0**



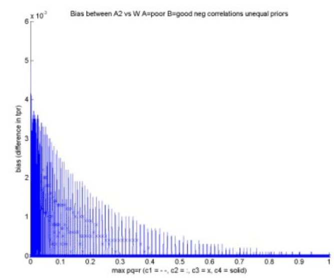

**A=fair B=poor *AND* ρ ≤ 0**



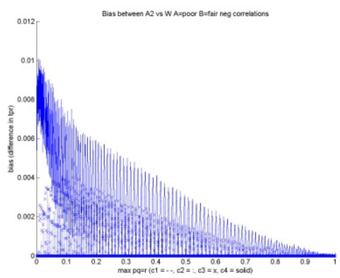

**A=fair B=poor *AND* unequal priors ρ ≤ 0**



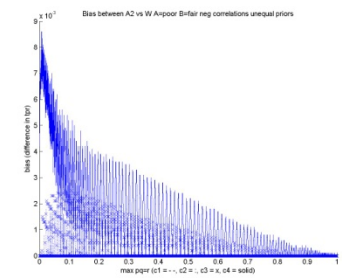

**A=poor B=good *AND* ρ ≤ 0**



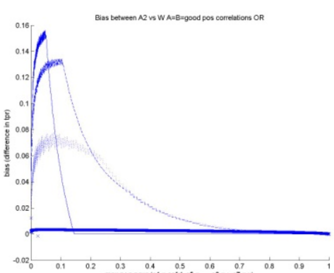

**A=poor B=good *AND* unequal priors ρ ≤ 0**



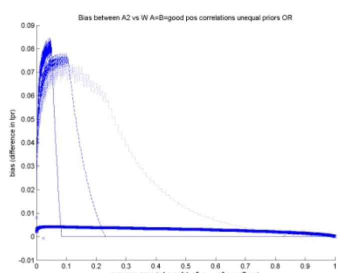

**A=poor B=fair *AND* ρ ≤ 0**



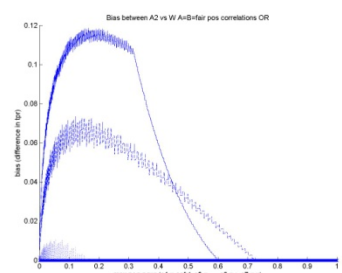

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



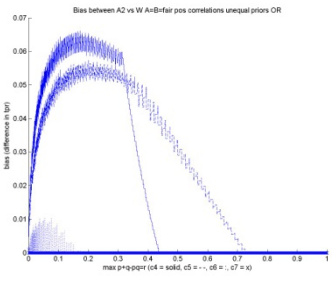

**A=B=good *OR* ρ ≥ 0**



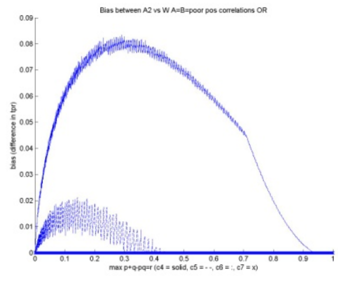

**A=B=good *OR* unequal priors ρ ≥ 0**



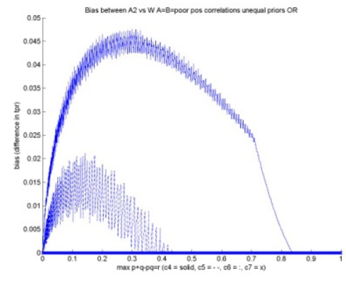

**A=B=fair *OR* ρ ≥ 0**



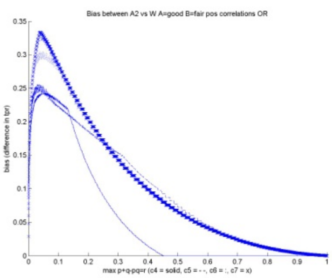

**A=B=fair *OR* unequal priors ρ ≥ 0**



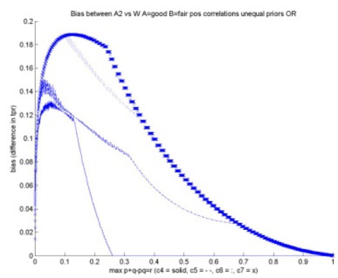

**A=B=poor *OR* ρ ≥ 0**



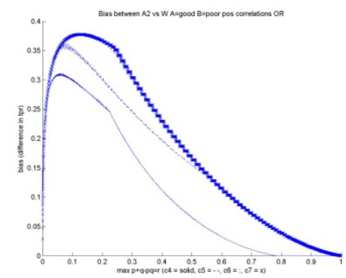

**A=B=poor unequal priors *OR* ρ ≥ 0**



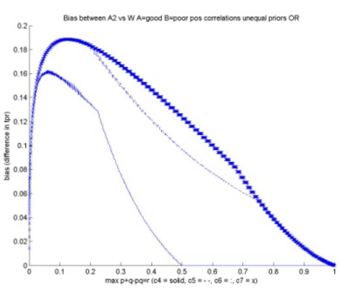

**A=good B=fair *OR* ρ ≥ 0**



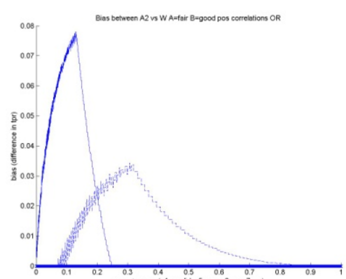

**A=good B=fair *OR* unequal priors ρ ≥ 0**



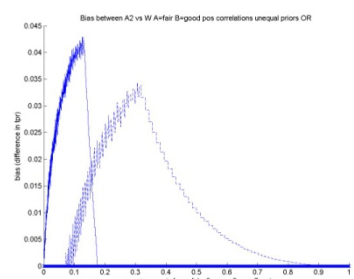

**A=good B=poor *OR* ρ ≥ 0**



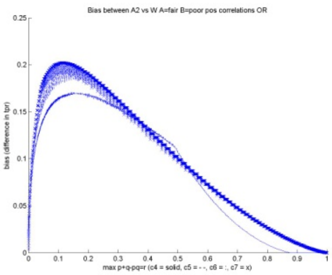

**A=good B=poor *OR* unequal priors ρ ≥ 0**



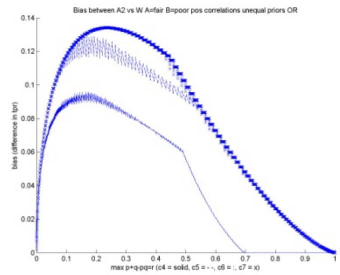

**A=fair B=good *OR* ρ ≥ 0**



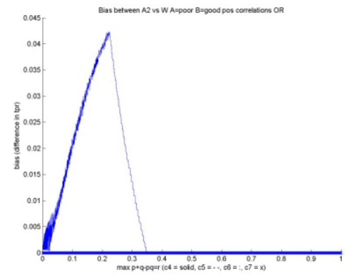

**A=fair B=good *OR* unequal priors ρ ≥ 0**



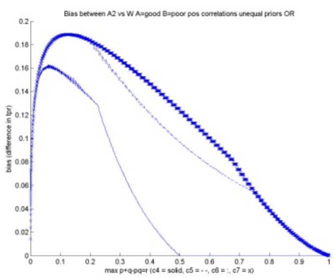

**A=fair B=poor *OR* ρ ≥ 0**



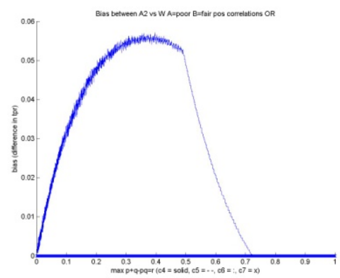

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



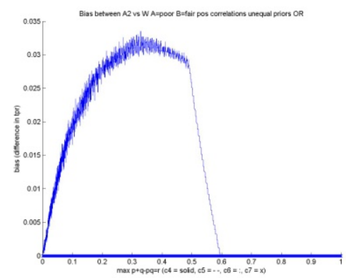

**A=poor B=good *OR* ρ ≥ 0**



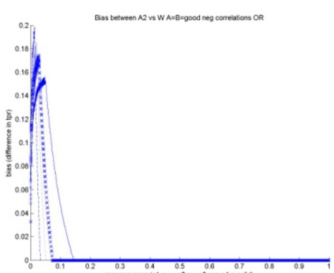

**A=poor B=good *OR* unequal priors ρ ≥ 0**



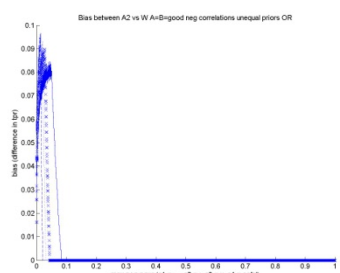

**A=poor B=fair *OR* ρ ≥ 0**



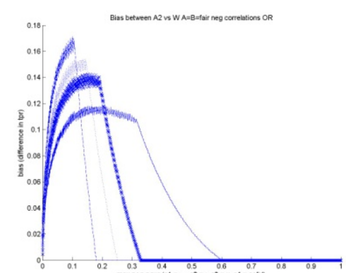

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



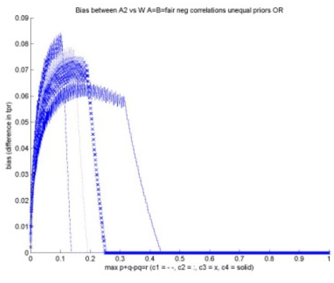

**A=B=good *OR* ρ ≤ 0**



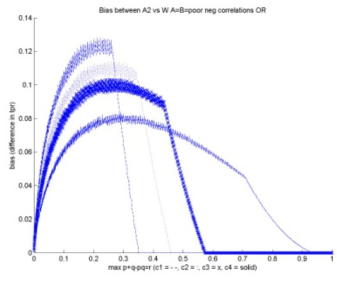

**A=B=good *OR* unequal priors ρ ≤ 0**



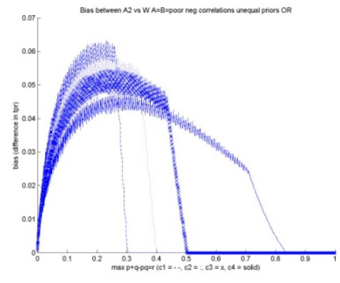

**A=B=fair *OR* ρ ≤ 0**



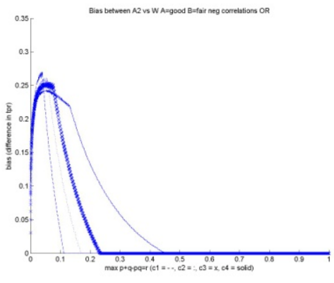

**A=B=fair *OR* unequal priors ρ ≤ 0**



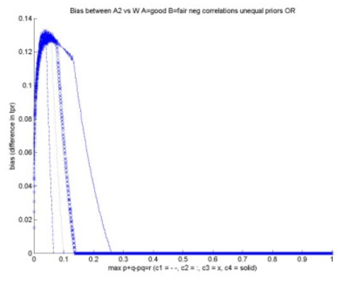

**A=B=poor *OR* ρ ≤ 0**



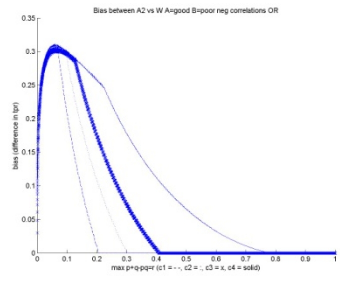

**A=B=poor unequal priors *OR* ρ ≤ 0**



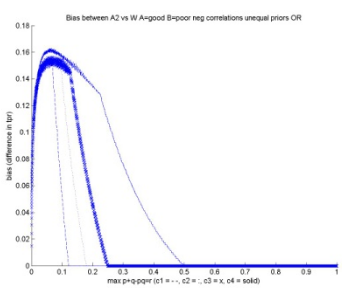

**A=good B=fair *OR* ρ ≤ 0**



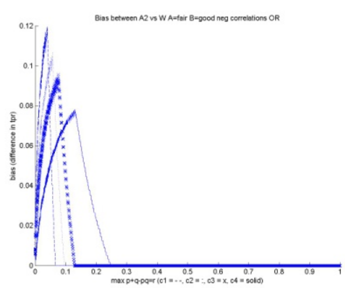

**A=good B=fair *OR* unequal priors ρ ≤ 0**



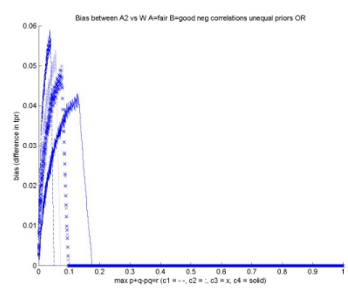

**A=good B=poor *OR* ρ ≤ 0**



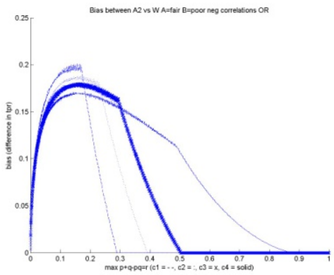

**A=good B=poor *OR* unequal priors ρ ≤ 0**



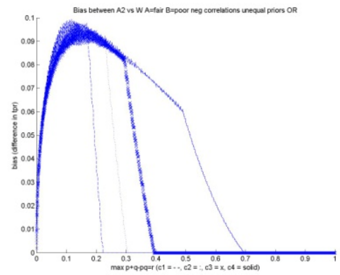

**A=fair B=good *OR* ρ ≤ 0**



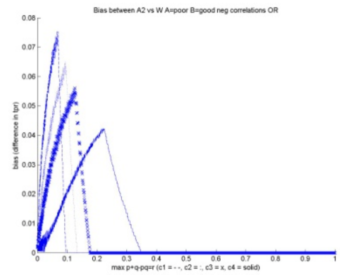

**A=fair B=good *OR* unequal priors ρ ≤ 0**



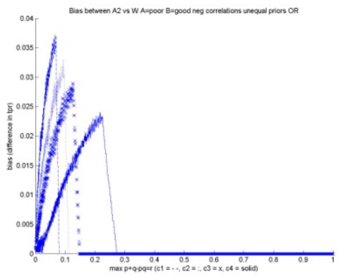
**A=fair B=poor *OR* ρ ≤ 0**



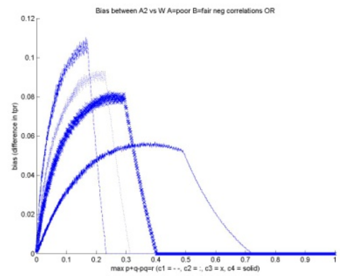
**A=fair B=poor *OR* unequal priors ρ ≤ 0**



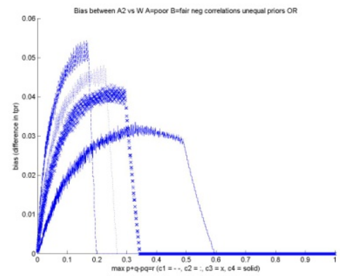
**A=poor B=good *OR* ρ ≤ 0**



**A=poor B=good *OR* unequal priors ρ ≤ 0**

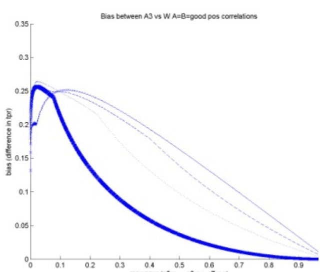


**A=poor B=fair *OR* ρ ≤ 0**

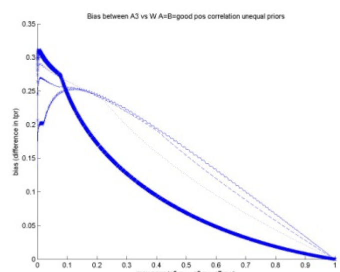


**A=poor B=fair *OR* unequal priors ρ ≤ 0**

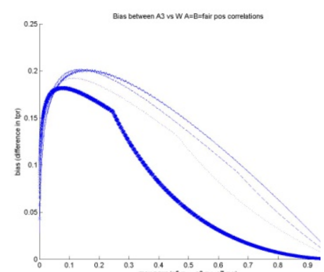
### *B.1.3 – Across III versus Within*



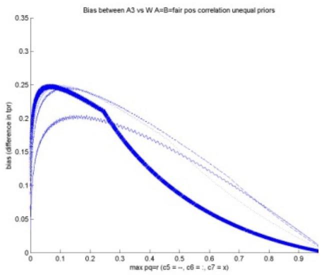
**A=B=good *AND* ρ ≥ 0**



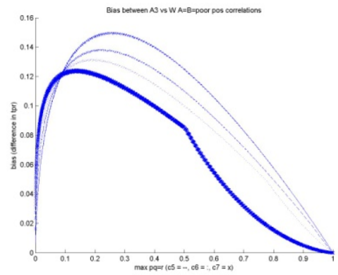
**A=B=good *AND* unequal priors ρ ≥ 0**



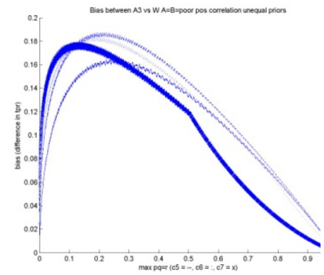
**A=B=fair *AND* ρ ≥ 0**



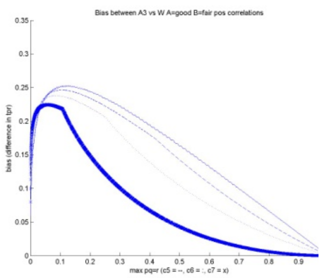

**A=B=fair *AND* unequal priors ρ ≥ 0**



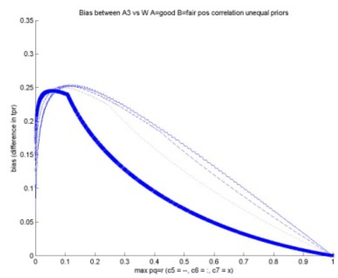

**A=B=poor *AND* ρ ≥ 0**



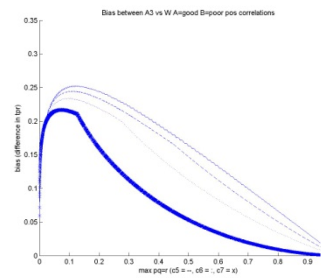

**A=B=poor unequal priors *AND* ρ ≥ 0**



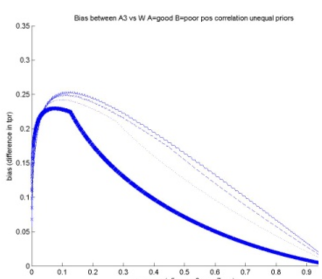

**A=good B=fair *AND* ρ ≥ 0**



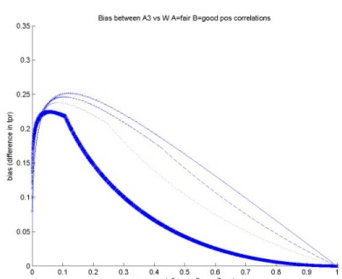

**A=good B=fair *AND* unequal priors ρ ≥ 0**



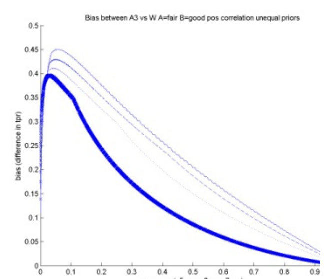

**A=good B=poor *AND* ρ ≥ 0**



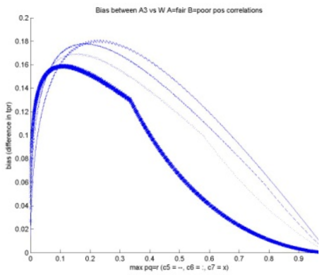

**A=good B=poor *AND* unequal priors ρ ≥ 0**



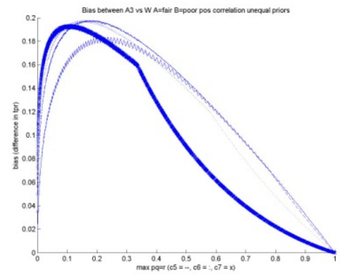

**A=fair B=good *AND* ρ ≥ 0**



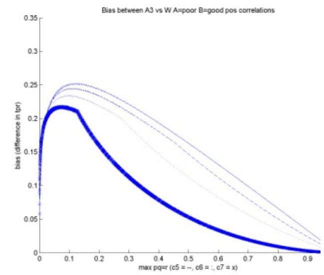

**A=fair B=good *AND* unequal priors ρ ≥ 0**



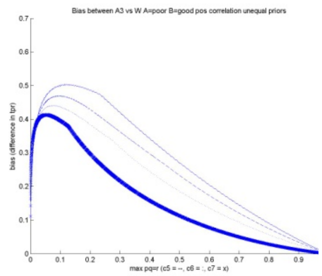
**A=fair B=poor *AND* ρ ≥ 0**



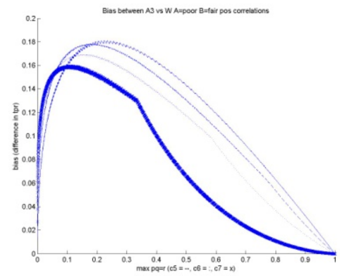
**A=fair B=poor *AND* unequal priors ρ ≥ 0**



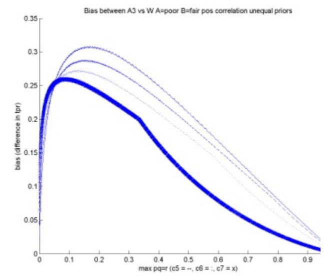
**A=poor B=good *AND* ρ ≥ 0**



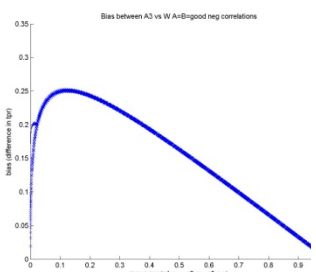
**A=poor B=good *AND* unequal priors ρ ≥ 0**



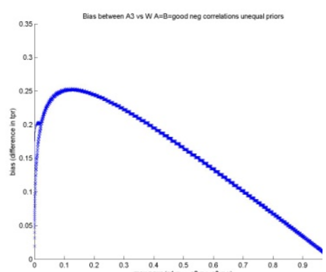
**A=poor B=fair *AND* ρ ≥ 0**



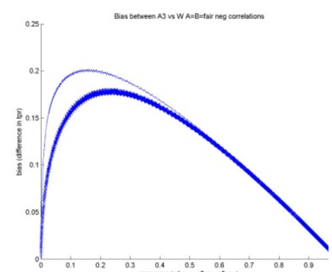
**A=poor B=fair *AND* unequal priors ρ ≥ 0**



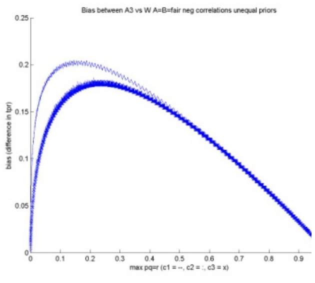
**A=B=good *AND* ρ ≤ 0**



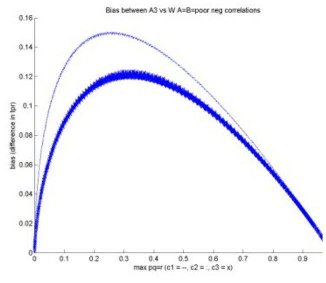
**A=B=good *AND* unequal priors ρ ≤ 0**



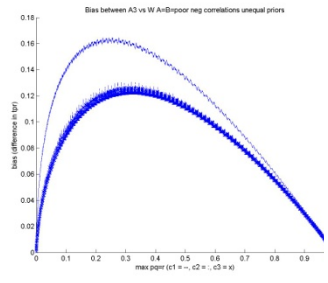
**A=B=fair *AND* ρ ≤ 0**



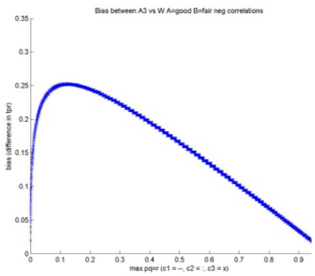

**A=B=fair *AND* unequal priors ρ ≤ 0**



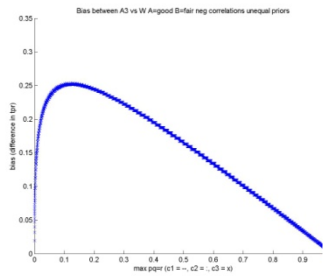

**A=B=poor *AND* ρ ≤ 0**



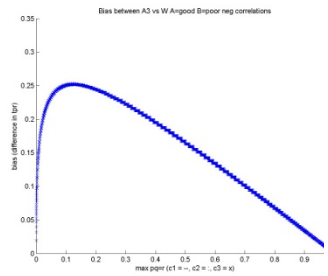

**A=B=poor unequal priors *AND* ρ ≤ 0**



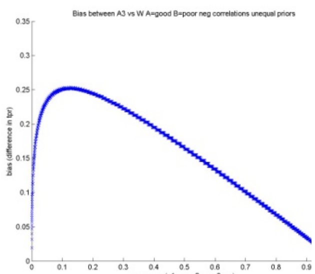

**A=good B=fair *AND* ρ ≤ 0**



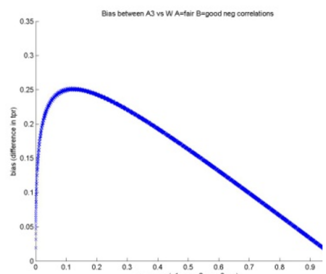

**A=good B=fair *AND* unequal priors ρ ≤ 0**



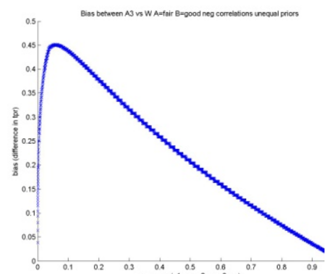

**A=good B=poor *AND* ρ ≤ 0**



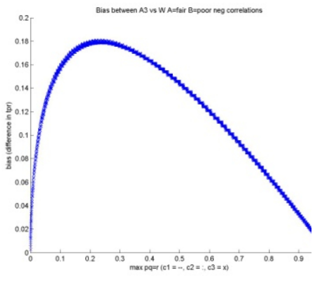

**A=good B=poor *AND* unequal priors ρ ≤ 0**



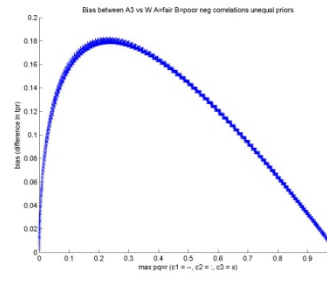

**A=fair B=good *AND* ρ ≤ 0**



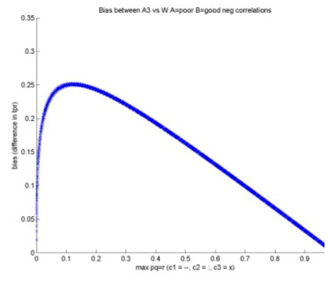

**A=fair B=good *AND* unequal priors ρ ≤ 0**



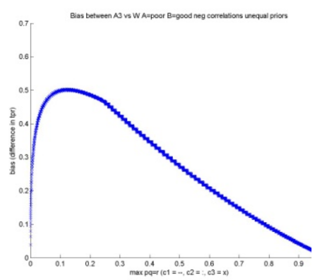

**A=fair B=poor *AND* ρ ≤ 0**



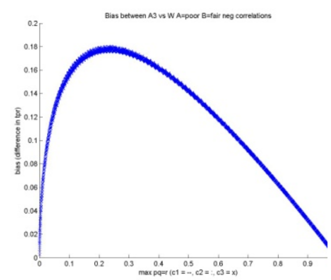

**A=fair B=poor *AND* unequal priors ρ ≤ 0**



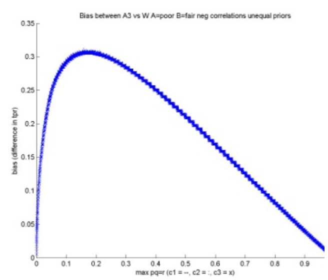

**A=poor B=good *AND* ρ ≤ 0**



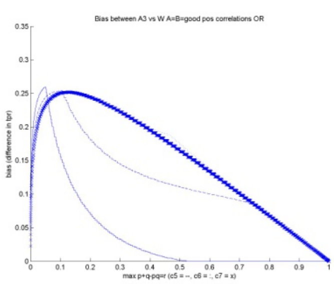

**A=poor B=good *AND* unequal priors ρ ≤ 0**



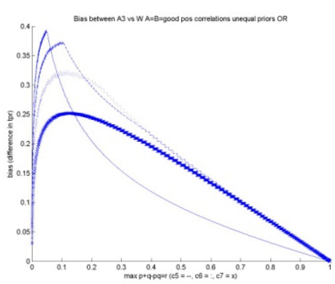

**A=poor B=fair *AND* ρ ≤ 0**



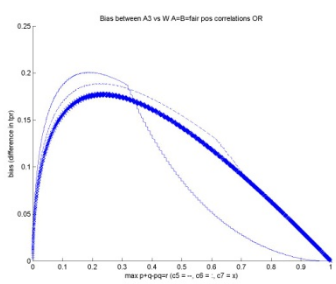

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



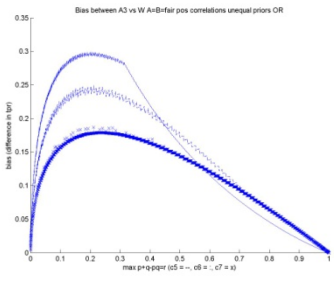

**A=B=good *OR* ρ ≥ 0**

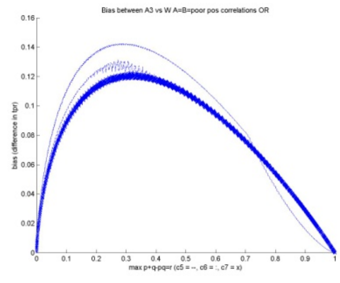


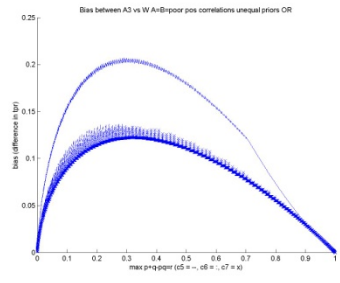

**A=B=good *OR* unequal priors ρ ≥ 0**




**A=B=fair *OR* ρ ≥ 0**



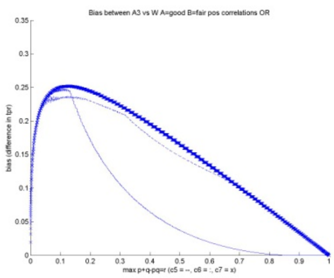



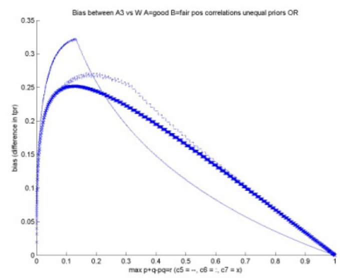



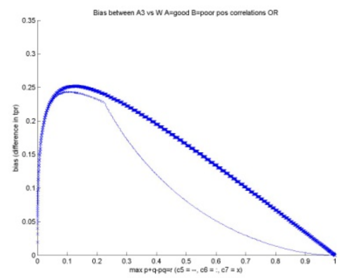

**A=B=fair *OR* unequal priors ρ ≥ 0**

**A=B=poor *OR* ρ ≥ 0**

**A=B=poor unequal priors *OR* ρ ≥ 0**



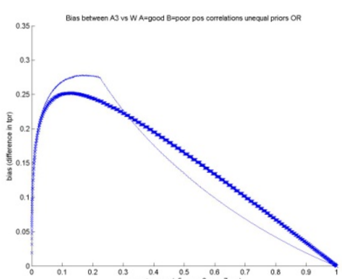



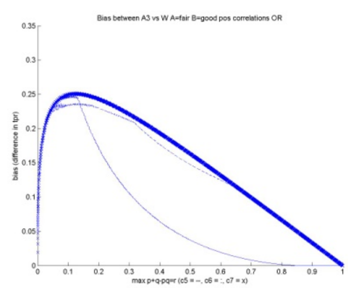



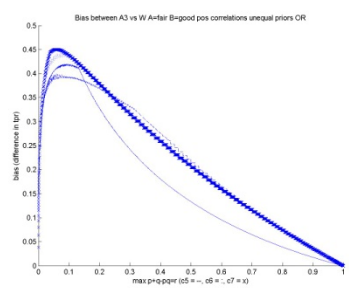

**A=good B=fair *OR* ρ ≥ 0**

**A=good B=fair *OR* unequal priors ρ ≥ 0**

**A=good B=poor *OR* ρ ≥ 0**



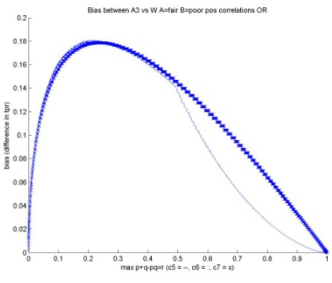



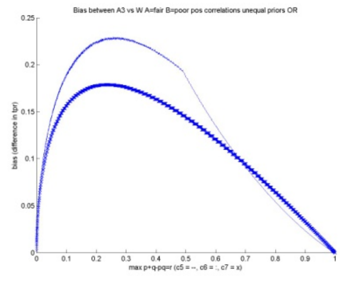

**A=good B=poor *OR* unequal priors ρ ≥ 0**

**A=fair B=good *OR* ρ ≥ 0**

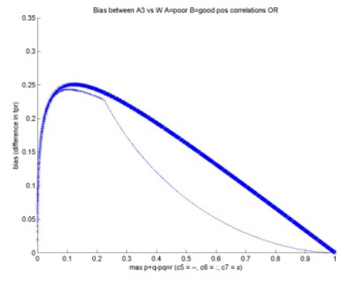
**A=fair B=good *OR* unequal priors ρ ≥ 0**



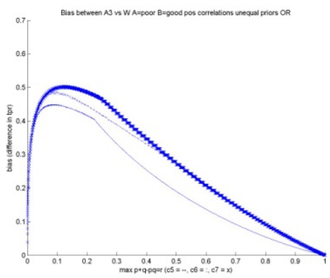

**A=fair B=poor *OR* ρ ≥ 0**



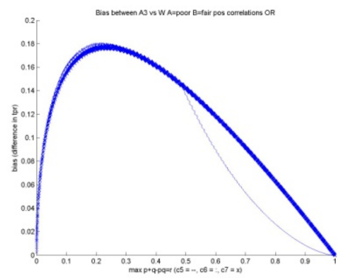

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



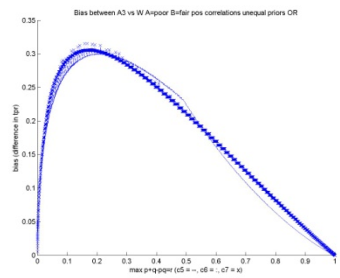

**A=poor B=good *OR* ρ ≥ 0**



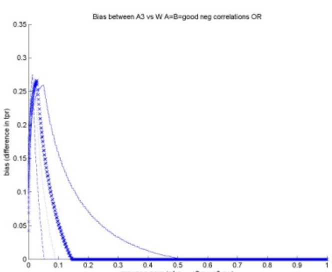

**A=poor B=good *OR* unequal priors ρ ≥ 0**



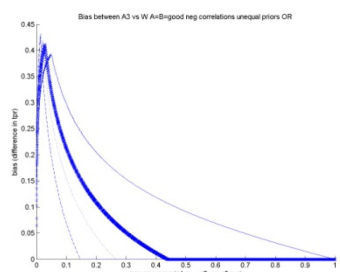

**A=poor B=fair *OR* ρ ≥ 0**



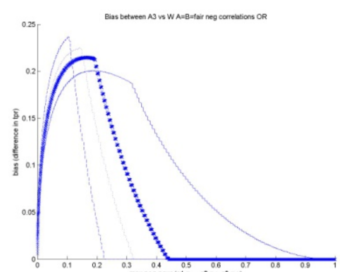

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



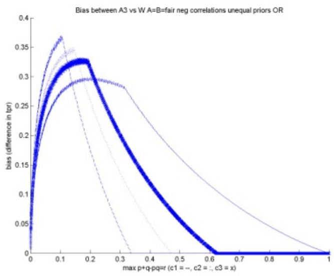

**A=B=good *OR* ρ ≤ 0**



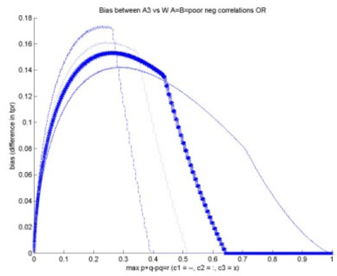

**A=B=good *OR* unequal priors ρ ≤ 0**



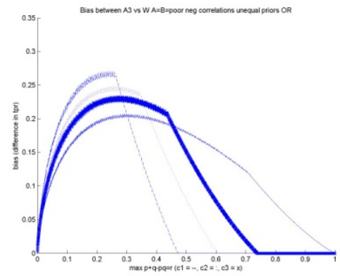

**A=B=fair *OR* ρ ≤ 0**



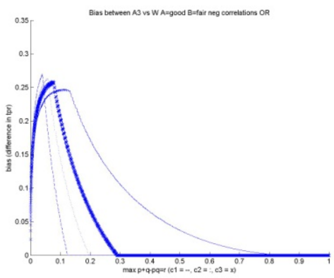

**A=B=fair *OR* unequal priors ρ ≤ 0**



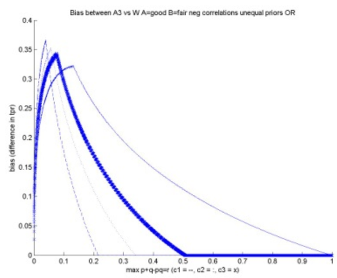

**A=B=poor *OR* ρ ≤ 0**



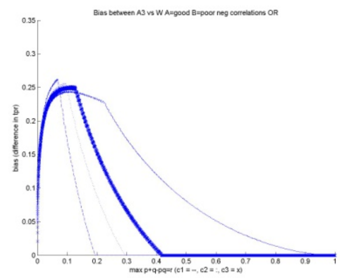

**A=B=poor unequal priors *OR* ρ ≤ 0**



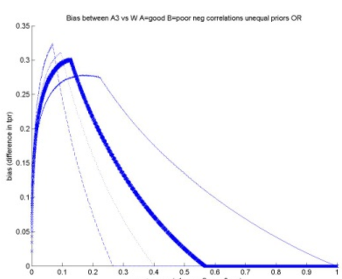

**A=good B=fair *OR* ρ ≤ 0**



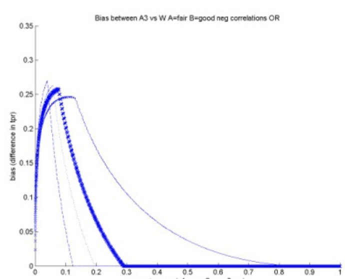

**A=good B=fair *OR* unequal priors ρ ≤ 0**



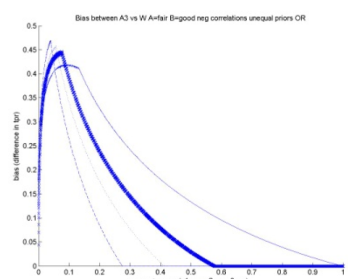

**A=good B=poor *OR* ρ ≤ 0**



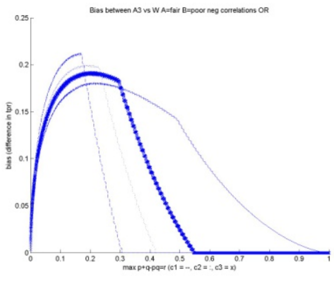

**A=good B=poor *OR* unequal priors ρ ≤ 0**



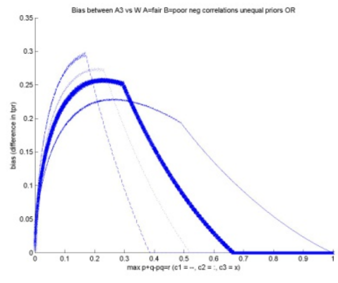

**A=fair B=good *OR* ρ ≤ 0**



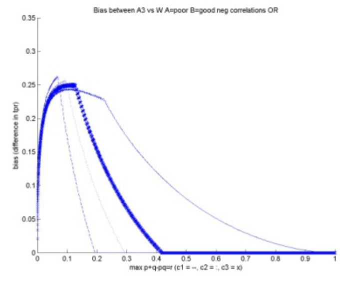

**A=fair B=good *OR* unequal priors ρ ≤ 0**



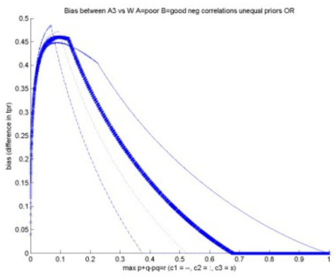

**A=fair B=poor *OR* ρ ≤ 0**



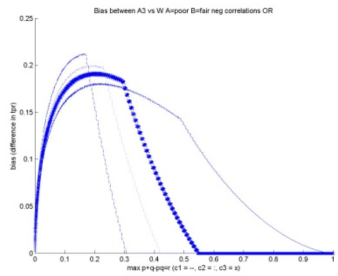

**A=fair B=poor *OR* unequal priors ρ ≤ 0**



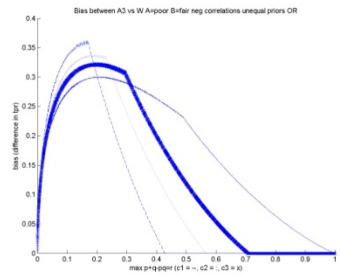

**A=poor B=good *OR* ρ ≤ 0**




**A=poor B=good *OR* unequal priors ρ ≤ 0**



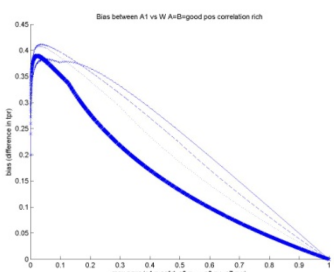

**A=poor B=fair *OR* ρ ≤ 0**



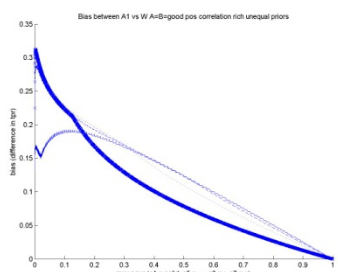

**A=poor B=fair *OR* unequal priors ρ ≤ 0**

## *B.2 – Rich Target Prevalence*

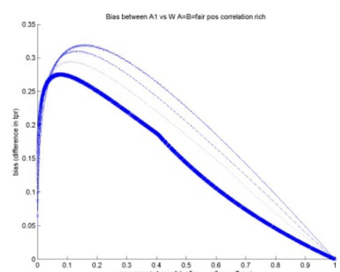
### *B.2.1 – Across I versus Within*



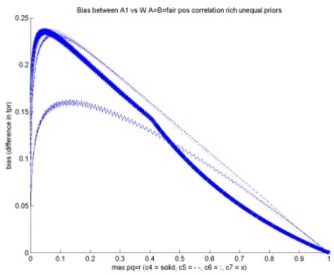

**A=B=good *AND* ρ ≥ 0**



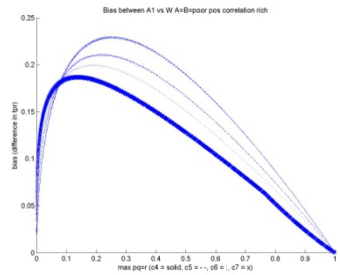

**A=B=good *AND* unequal priors ρ ≥ 0**



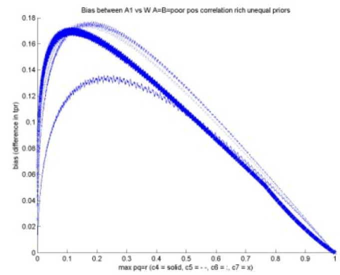

**A=B=fair *AND* ρ ≥ 0**



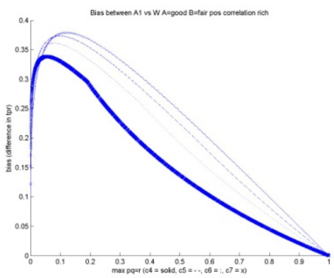

**A=B=fair *AND* unequal priors ρ ≥ 0**



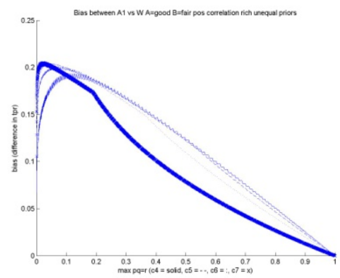

**A=B=poor *AND* ρ ≥ 0**



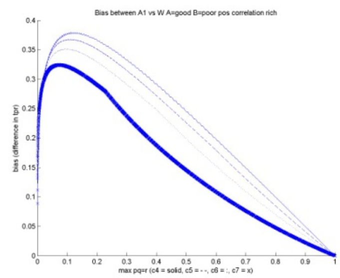

**A=B=poor unequal priors *AND* ρ ≥ 0**



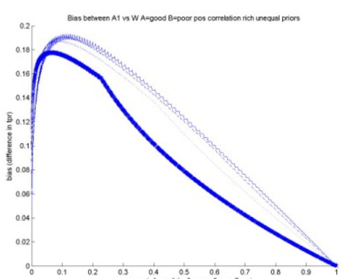

**A=good B=fair *AND* ρ ≥ 0**



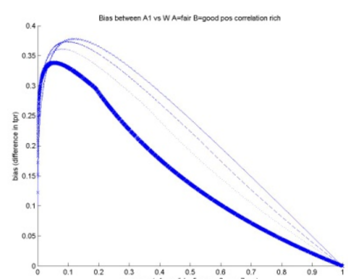

**A=good B=fair *AND* unequal priors ρ ≥ 0**



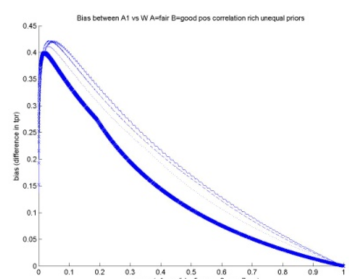

**A=good B=poor *AND* ρ ≥ 0**



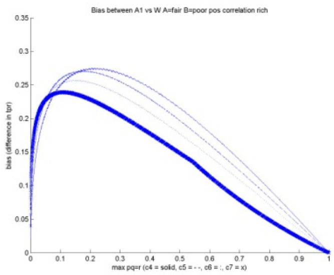

**A=good B=poor *AND* unequal priors ρ ≥ 0**



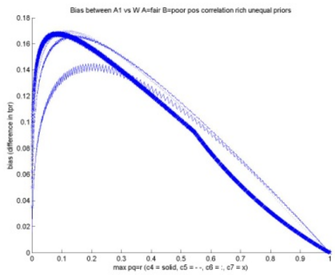

**A=fair B=good *AND* ρ ≥ 0**



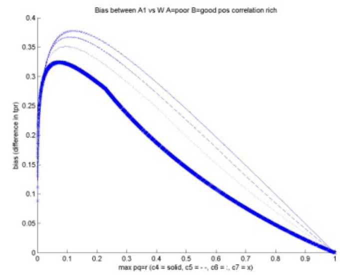

**A=fair B=good *AND* unequal priors ρ ≥ 0**



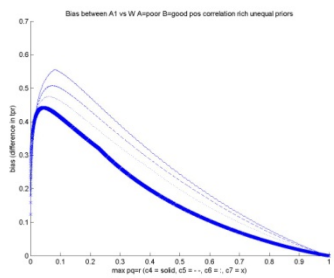

**A=fair B=poor *AND* ρ ≥ 0**



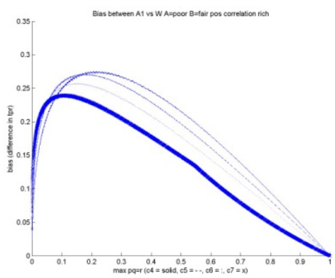

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



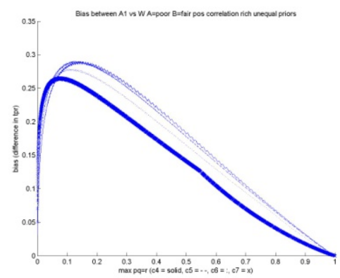

**A=poor B=good *AND* ρ ≥ 0**



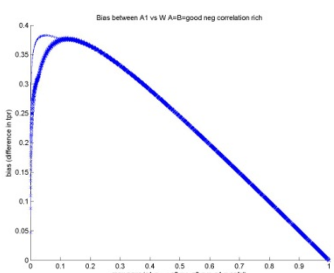

**A=poor B=good *AND* unequal priors ρ ≥ 0**



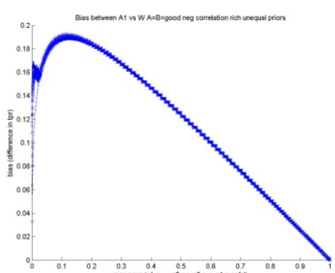

**A=poor B=fair *AND* ρ ≥ 0**



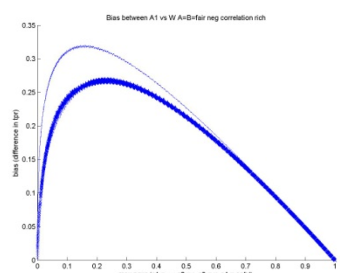

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



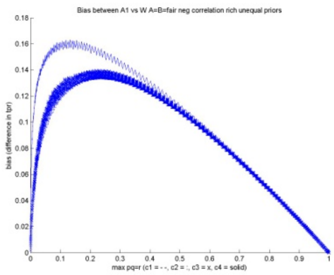

**A=B=good *AND* ρ ≤ 0**



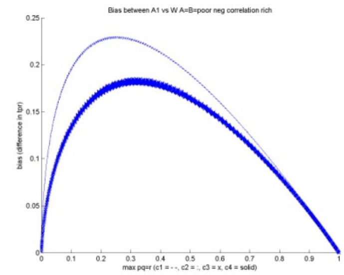

**A=B=good *AND* unequal priors ρ ≤ 0**



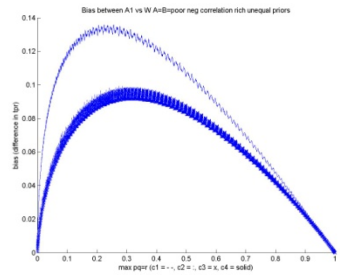

**A=B=fair *AND* ρ ≤ 0**



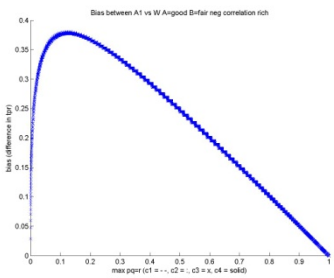

**A=B=fair *AND* unequal priors ρ ≤ 0**



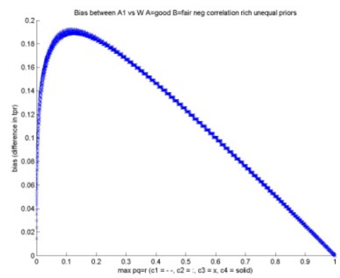

**A=B=poor *AND* ρ ≤ 0**



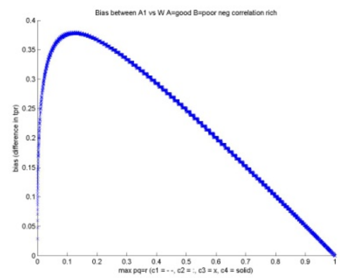

**A=B=poor unequal priors *AND* ρ ≤ 0**



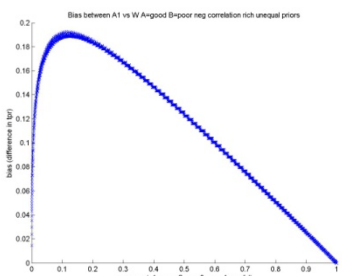

**A=good B=fair *AND* ρ ≤ 0**



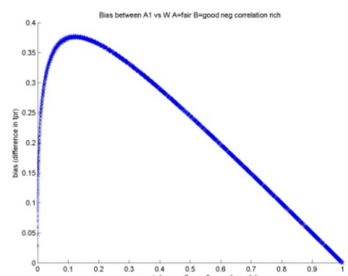

**A=good B=fair *AND* unequal priors ρ ≤ 0**



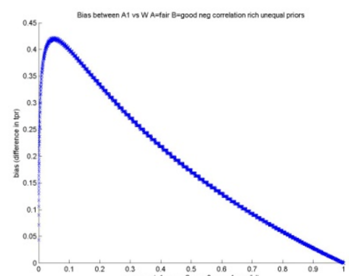

**A=good B=poor *AND* ρ ≤ 0**



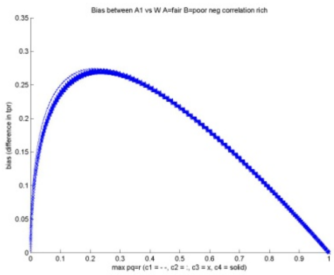

**A=good B=poor *AND* unequal priors ρ ≤ 0**

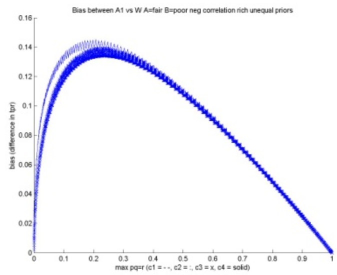


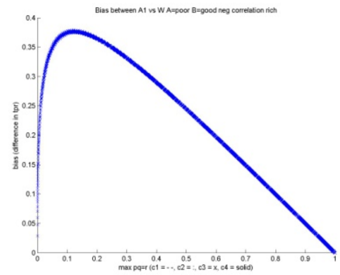

**A=fair B=good *AND* ρ ≤ 0**




**A=fair B=good *AND* unequal priors ρ ≤ 0**



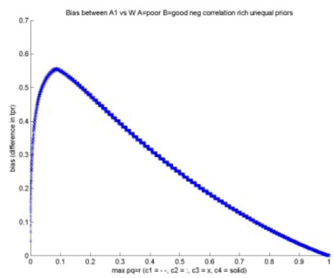



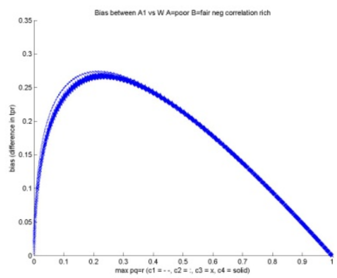



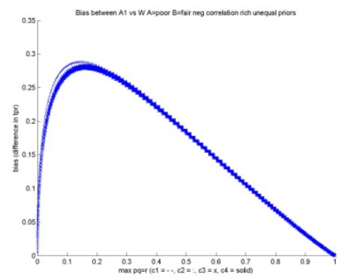

**A=fair B=poor *AND* ρ ≤ 0**

**A=fair B=poor *AND* unequal priors ρ ≤ 0**

**A=poor B=good *AND* ρ ≤ 0**



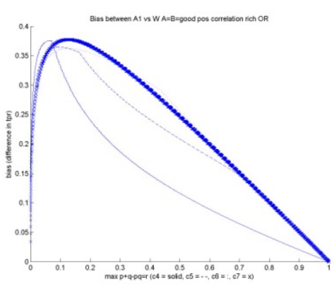



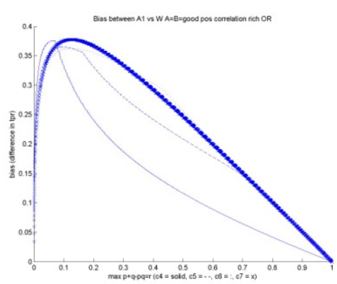



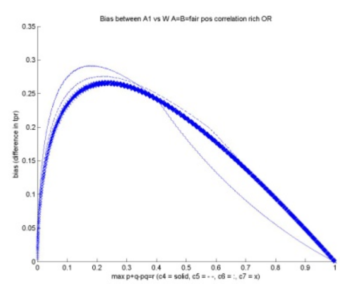

**A=poor B=good *AND* unequal priors ρ ≤ 0**

**A=poor B=fair *AND* ρ ≤ 0**

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



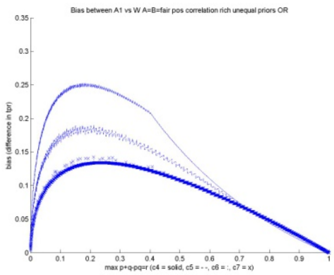



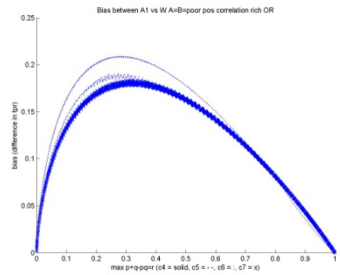

**A=B=good *OR* ρ ≥ 0**

**A=B=good *OR* unequal priors ρ ≥ 0**

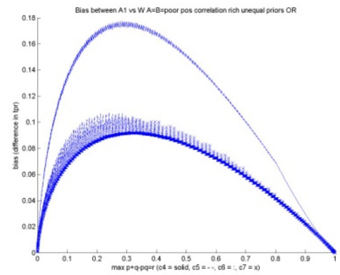
**A=B=fair *OR* ρ ≥ 0**



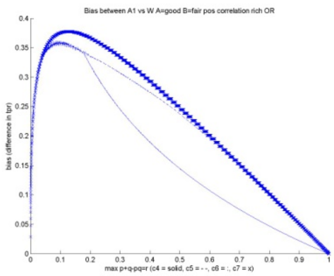

**A=B=fair *OR* unequal priors ρ ≥ 0**



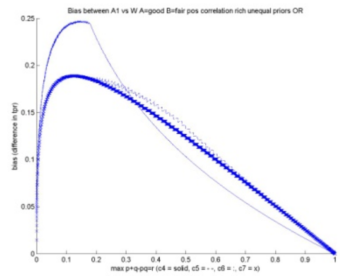

**A=B=poor *OR* ρ ≥ 0**



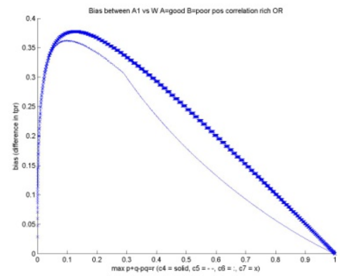

**A=B=poor unequal priors *OR* ρ ≥ 0**



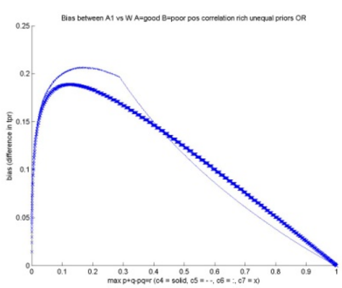

**A=good B=fair *OR* ρ ≥ 0**



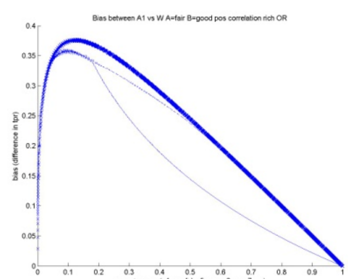

**A=good B=fair *OR* unequal priors ρ ≥ 0**



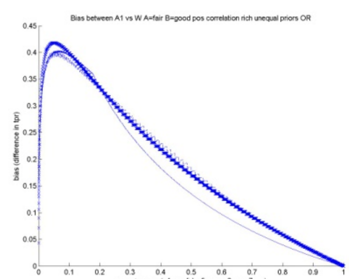

**A=good B=poor *OR* ρ ≥ 0**



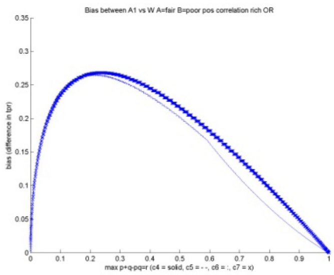

**A=good B=poor *OR* unequal priors ρ ≥ 0**



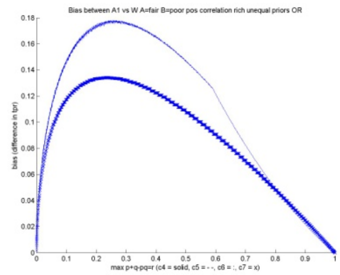

**A=fair B=good *OR* ρ ≥ 0**



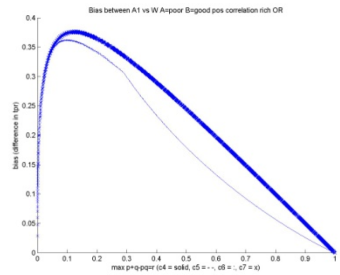

**A=fair B=good *OR* unequal priors ρ ≥ 0**



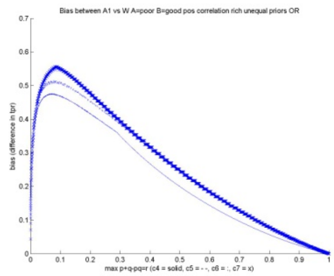

**A=fair B=poor *OR* ρ ≥ 0**



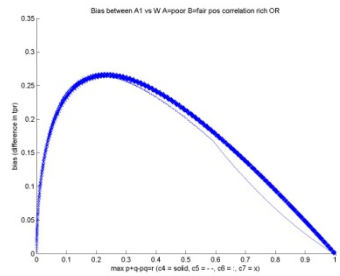

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



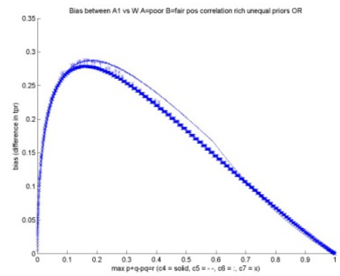

**A=poor B=good *OR* ρ ≥ 0**



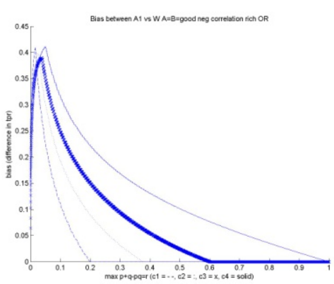

**A=poor B=good *OR* unequal priors ρ ≥ 0**



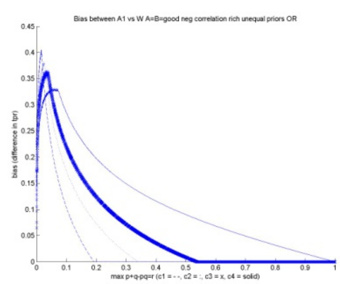

**A=poor B=fair *OR* ρ ≥ 0**



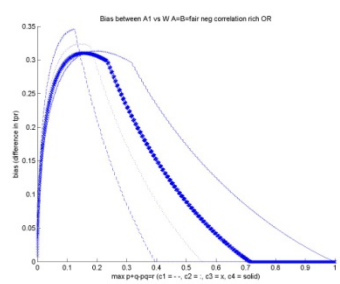

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



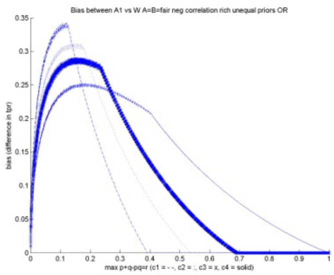

**A=B=good *OR* ρ ≤ 0**



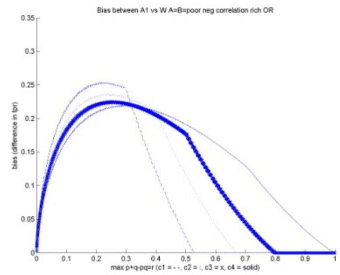

**A=B=good *OR* unequal priors ρ ≤ 0**



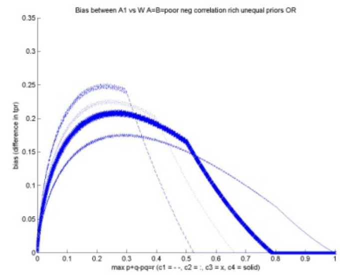

**A=B=fair *OR* ρ ≤ 0**



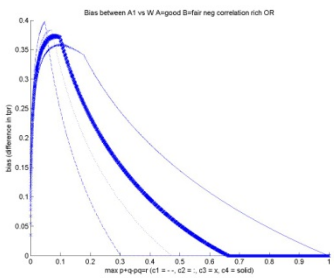

**A=B=fair *OR* unequal priors ρ ≤ 0**



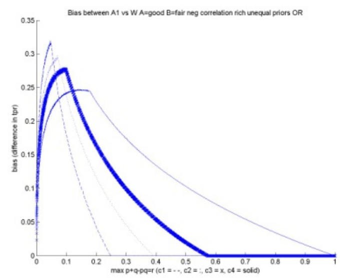

**A=B=poor *OR* ρ ≤ 0**



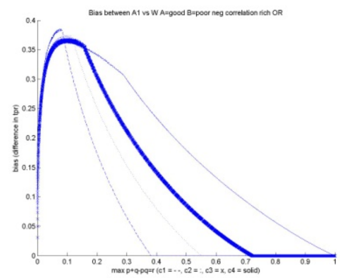

**A=B=poor unequal priors *OR* ρ ≤ 0**



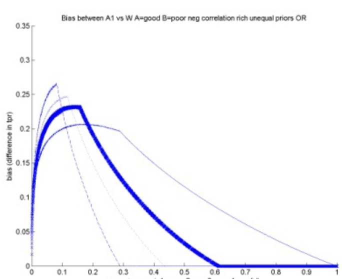

**A=good B=fair *OR* ρ ≤ 0**



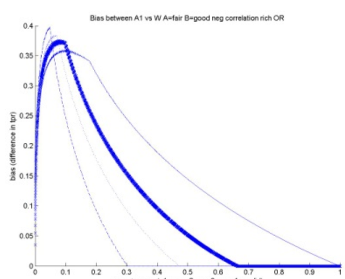

**A=good B=fair *OR* unequal priors ρ ≤ 0**



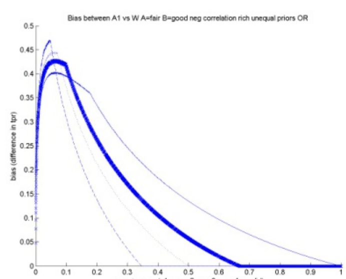

**A=good B=poor *OR* ρ ≤ 0**



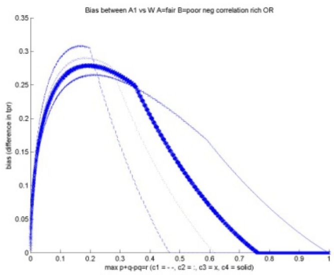

**A=good B=poor *OR* unequal priors ρ ≤ 0**



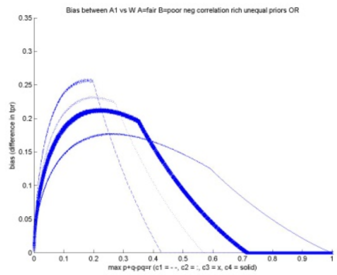

**A=fair B=good *OR* ρ ≤ 0**



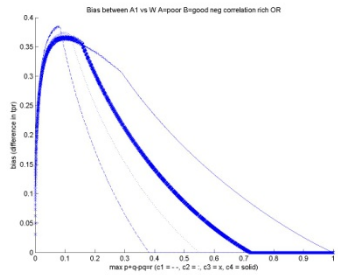

**A=fair B=good *OR* unequal priors ρ ≤ 0**



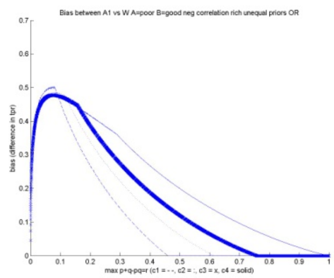

**A=fair B=poor *OR* ρ ≤ 0**



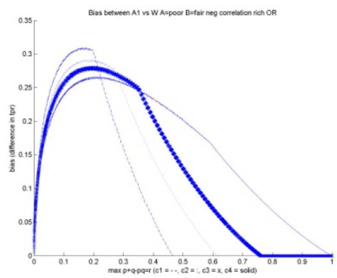

**A=fair B=poor *OR* unequal priors ρ ≤ 0**



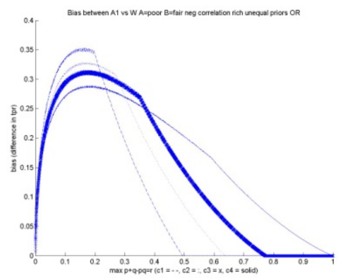

**A=poor B=good *OR* ρ ≤ 0**




**A=poor B=good *OR* unequal priors ρ ≤ 0**

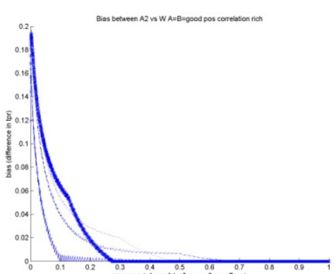



**A=poor B=fair *OR* ρ ≤ 0**

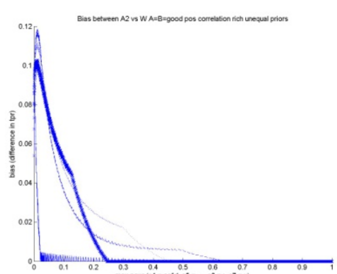



**A=poor B=fair *OR* unequal priors ρ ≤ 0**

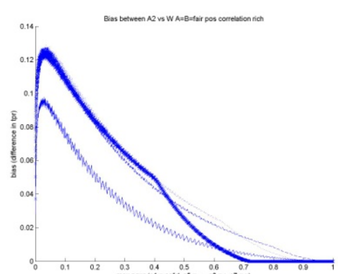
### *B.2.2 – Across II versus Within*



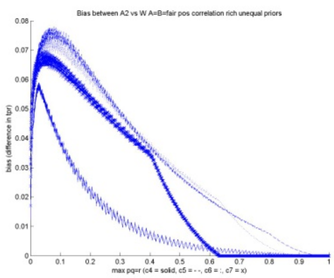

**A=B=good *AND* ρ ≥ 0**



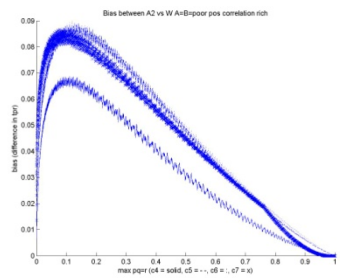

**A=B=good *AND* unequal priors ρ ≥ 0**



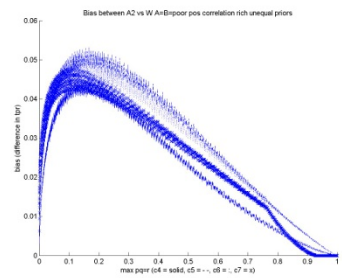

**A=B=fair *AND* ρ ≥ 0**



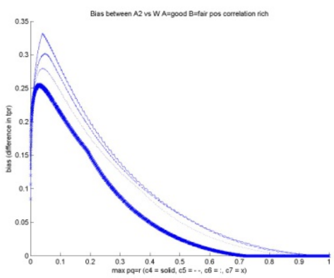

**A=B=fair *AND* unequal priors ρ ≥ 0**



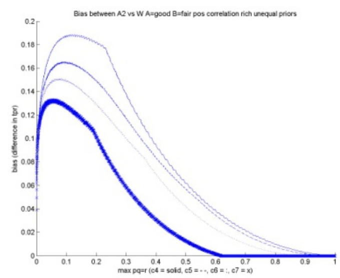

**A=B=poor *AND* ρ ≥ 0**



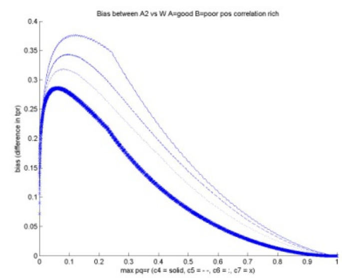

**A=B=poor unequal priors *AND* ρ ≥ 0**



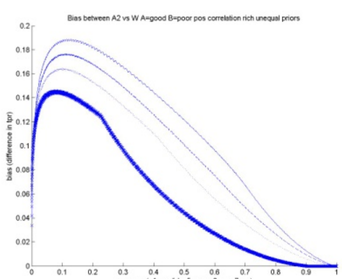

**A=good B=fair *AND* ρ ≥ 0**



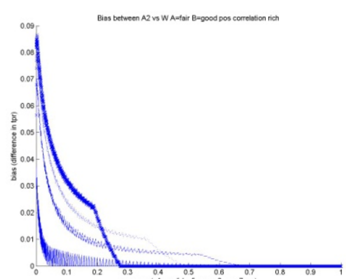

**A=good B=fair *AND* unequal priors ρ ≥ 0**



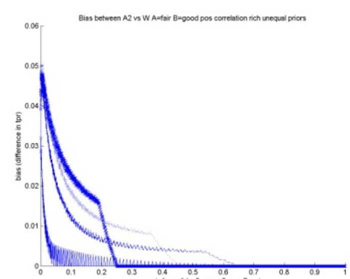

**A=good B=poor *AND* ρ ≥ 0**



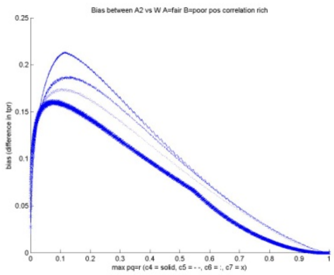

**A=good B=poor *AND* unequal priors ρ ≥ 0**



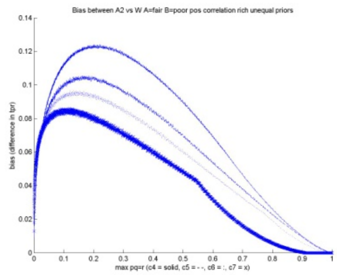

**A=fair B=good *AND* ρ ≥ 0**



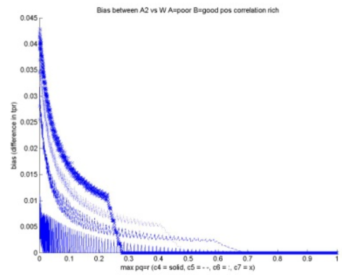

**A=fair B=good *AND* unequal priors ρ ≥ 0**



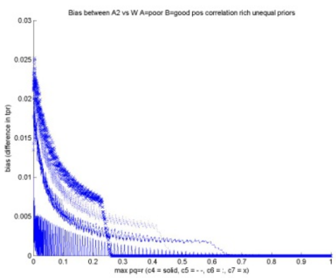

**A=fair B=poor *AND* ρ ≥ 0**



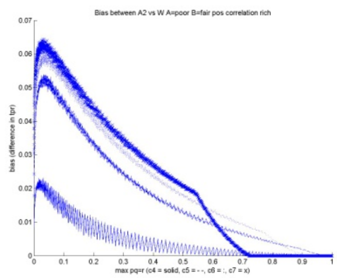

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



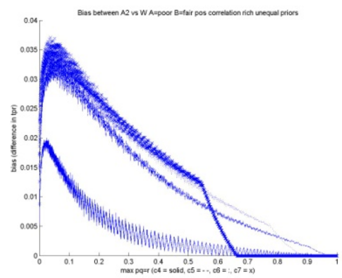

**A=poor B=good *AND* ρ ≥ 0**



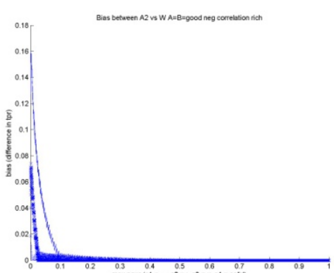

**A=poor B=good *AND* unequal priors ρ ≥ 0**



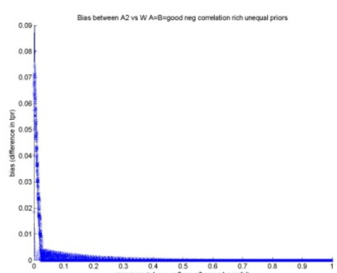

**A=poor B=fair *AND* ρ ≥ 0**



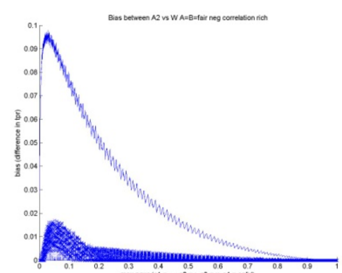

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



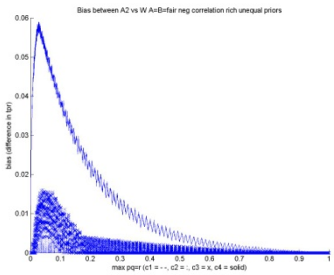

**A=B=good *AND* ρ ≤ 0**



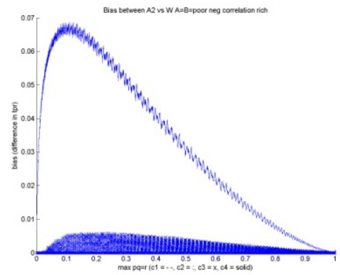

**A=B=good *AND* unequal priors ρ ≤ 0**



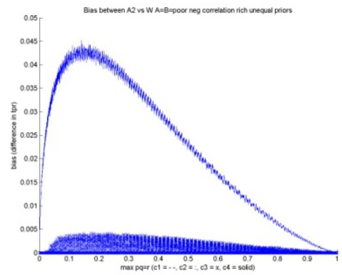

**A=B=fair *AND* ρ ≤ 0**



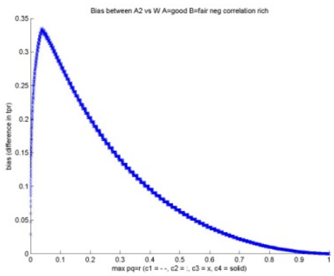

**A=B=fair *AND* unequal priors ρ ≤ 0**



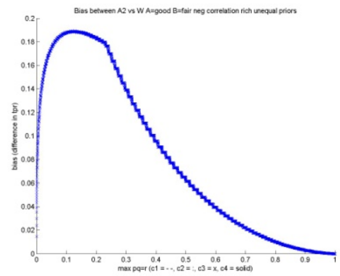

**A=B=poor *AND* ρ ≤ 0**



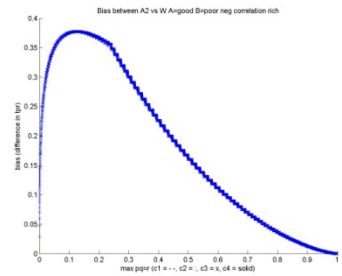

**A=B=poor unequal priors *AND* ρ ≤ 0**



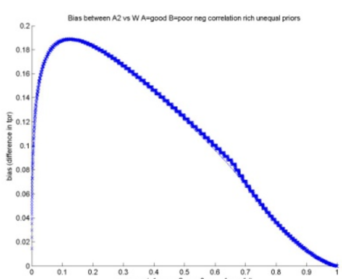

**A=good B=fair *AND* ρ ≤ 0**



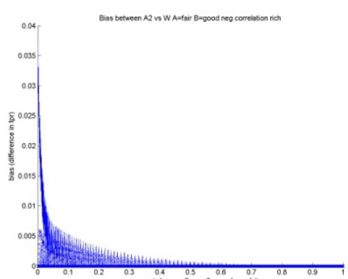

**A=good B=fair *AND* unequal priors ρ ≤ 0**



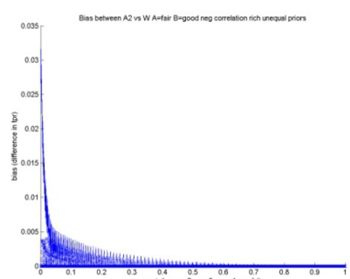

**A=good B=poor *AND* ρ ≤ 0**



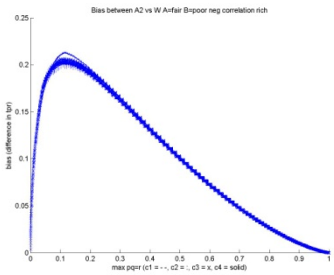

**A=good B=poor *AND* unequal priors ρ ≤ 0**

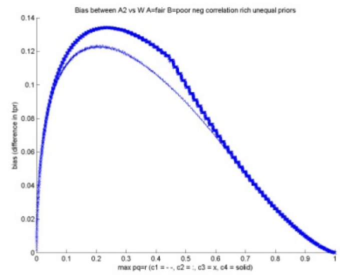


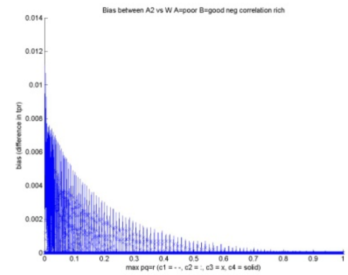

**A=fair B=good *AND* ρ ≤ 0**




**A=fair B=good *AND* unequal priors ρ ≤ 0**



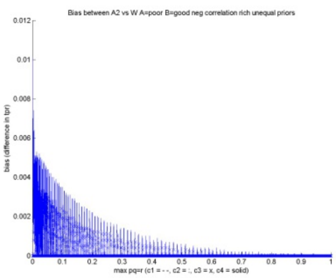



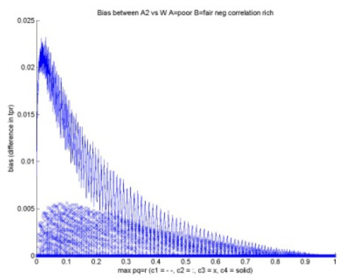



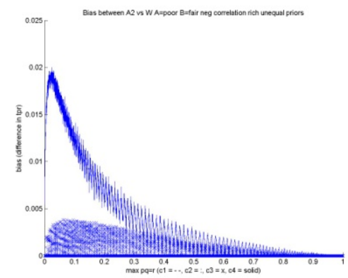

**A=fair B=poor *AND* ρ ≤ 0**

**A=fair B=poor *AND* unequal priors ρ ≤ 0**

**A=poor B=good *AND* ρ ≤ 0**



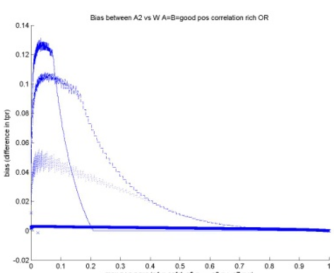



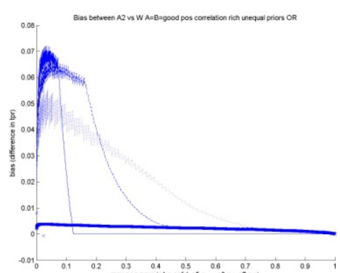



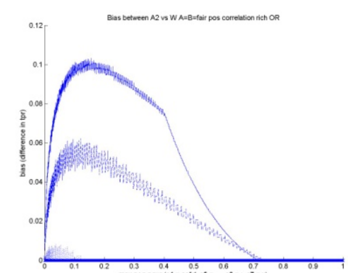

**A=poor B=good *AND* unequal priors ρ ≤ 0**

**A=poor B=fair *AND* ρ ≤ 0**

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



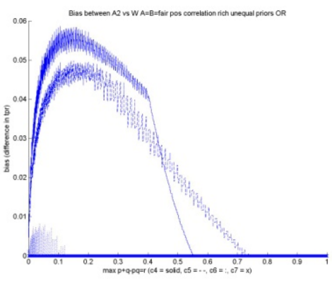



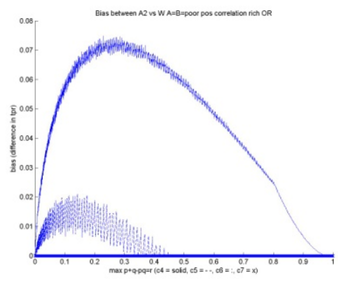

**A=B=good *OR* ρ ≥ 0**

**A=B=good *OR* unequal priors ρ ≥ 0**

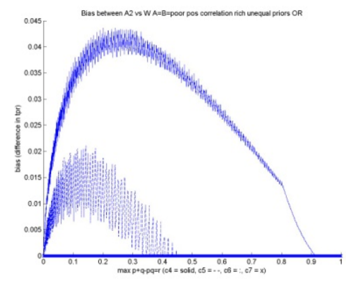
**A=B=fair *OR* ρ ≥ 0**



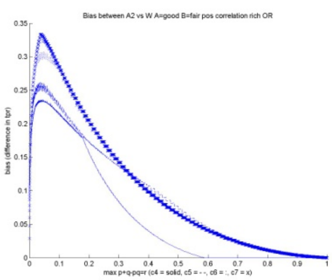

A=B=fair *OR* unequal priors ρ ≥ 0



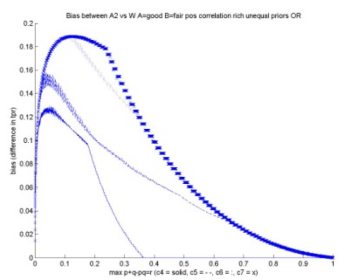

A=B=poor *OR* ρ ≥ 0



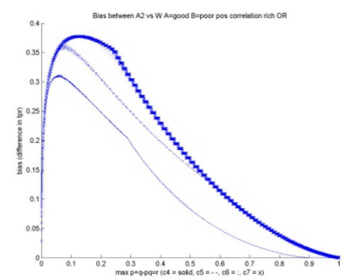

A=B=poor unequal priors *OR* ρ ≥ 0



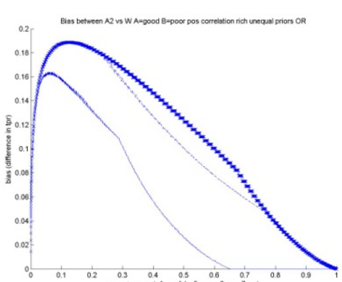

A=good B=fair *OR* ρ ≥ 0



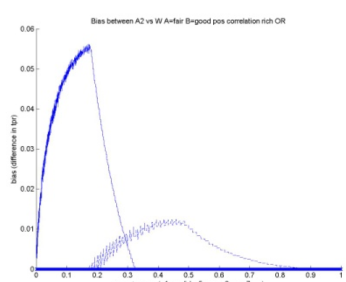

A=good B=fair *OR* unequal priors ρ ≥ 0



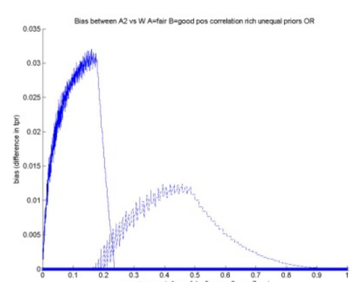

A=good B=poor *OR* ρ ≥ 0



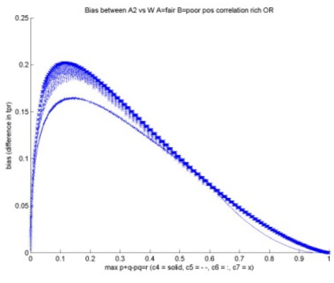

A=good B=poor *OR* unequal priors ρ ≥ 0



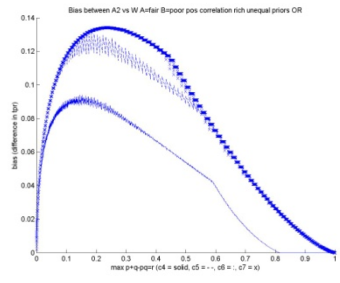

A=fair B=good *OR* ρ ≥ 0



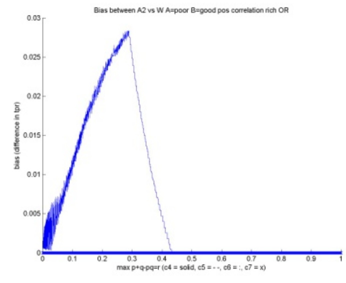

A=fair B=good *OR* unequal priors ρ ≥ 0



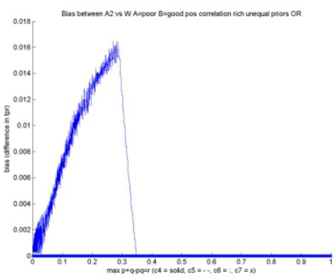

**A=fair B=poor *OR* ρ ≥ 0**



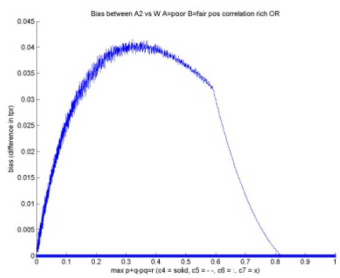

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



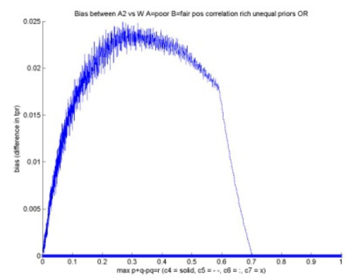

**A=poor B=good *OR* ρ ≥ 0**



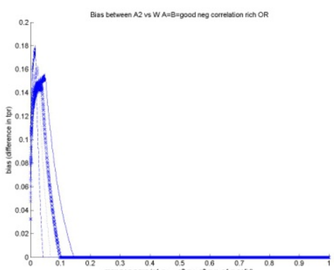

**A=poor B=good *OR* unequal priors ρ ≥ 0**



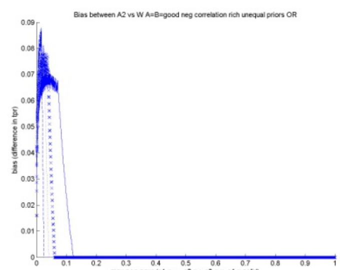

**A=poor B=fair *OR* ρ ≥ 0**



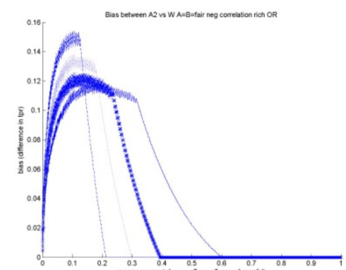

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



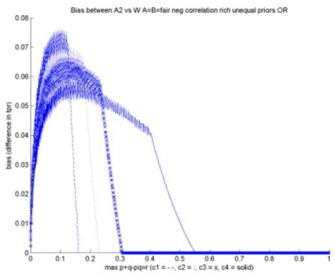

**A=B=good *OR* ρ ≤ 0**



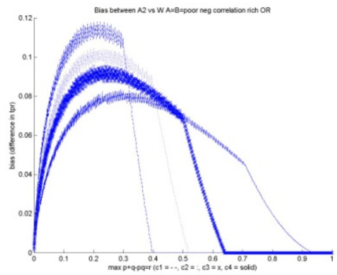

**A=B=good *OR* unequal priors ρ ≤ 0**



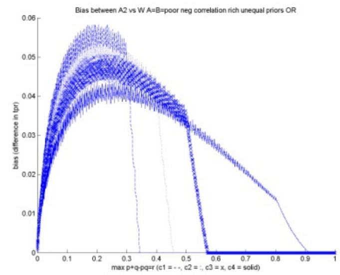

**A=B=fair *OR* ρ ≤ 0**



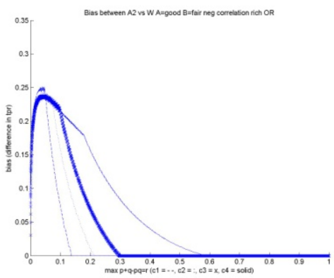

**A=B=fair *OR* unequal priors ρ ≤ 0**



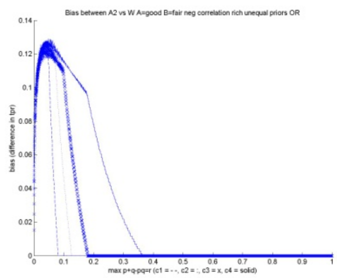

**A=B=poor *OR* ρ ≤ 0**



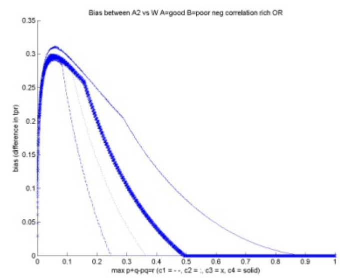

**A=B=poor unequal priors *OR* ρ ≤ 0**



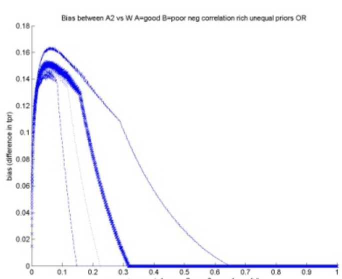

**A=good B=fair *OR* ρ ≤ 0**



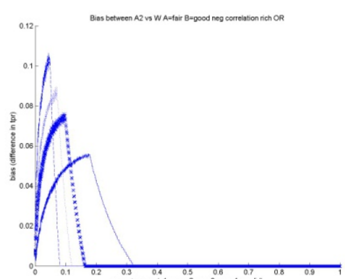

**A=good B=fair *OR* unequal priors ρ ≤ 0**



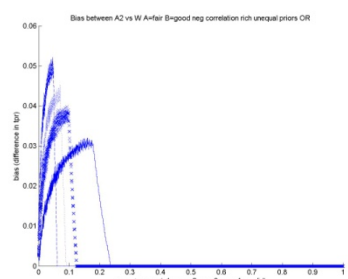

**A=good B=poor *OR* ρ ≤ 0**



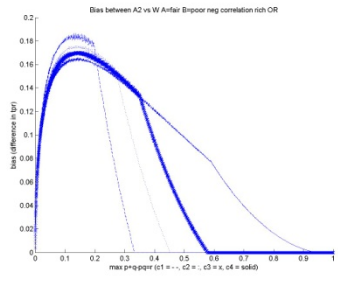

**A=good B=poor *OR* unequal priors ρ ≤ 0**



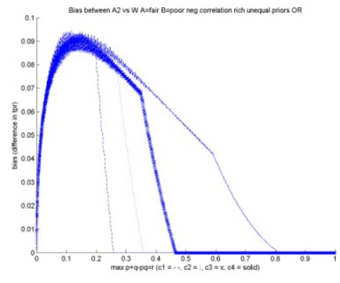

**A=fair B=good *OR* ρ ≤ 0**



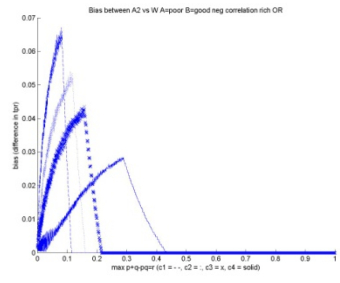

**A=fair B=good *OR* unequal priors ρ ≤ 0**



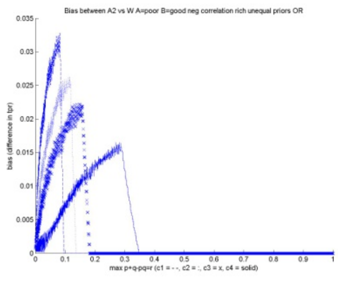
**A=fair B=poor *OR* ρ ≤ 0**



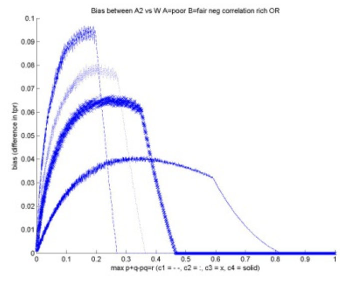
**A=fair B=poor *OR* unequal priors ρ ≤ 0**



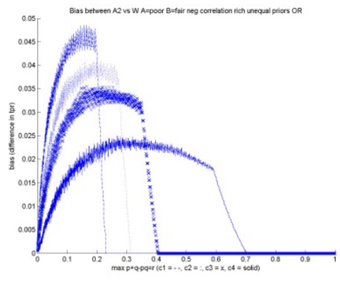
**A=poor B=good *OR* ρ ≤ 0**



**A=poor B=good *OR* unequal priors ρ ≤ 0**

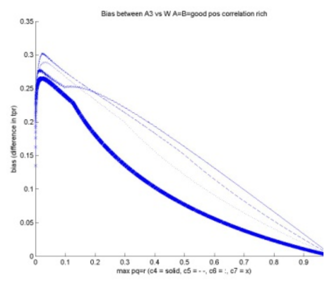


**A=poor B=fair *OR* ρ ≤ 0**

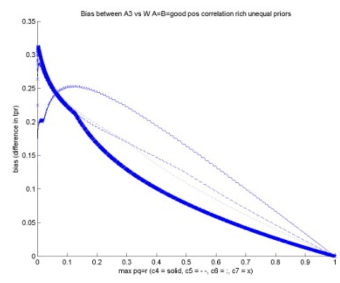


**A=poor B=fair *OR* unequal priors ρ ≤ 0**

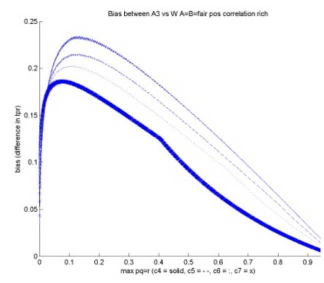
### *B.2.3 – Across III versus Within*



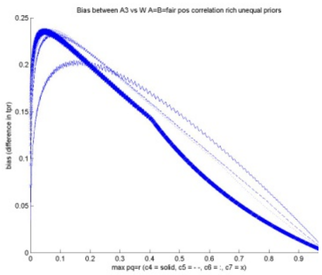
**A=B=good *AND* ρ ≥ 0**



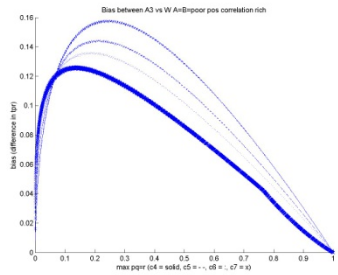
**A=B=good *AND* unequal priors ρ ≥ 0**



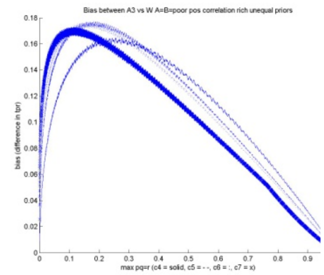
**A=B=fair *AND* ρ ≥ 0**



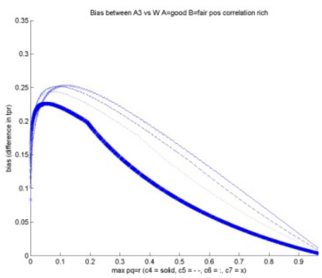

**A=B=fair *AND* unequal priors ρ ≥ 0**



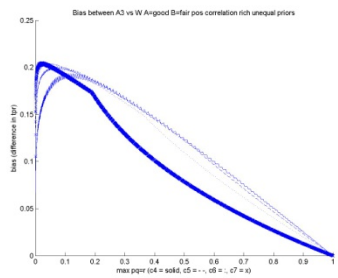

**A=B=poor *AND* ρ ≥ 0**



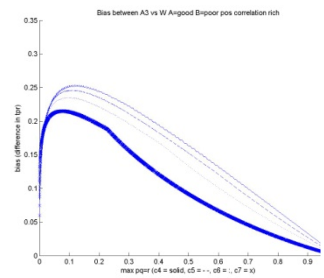

**A=B=poor unequal priors *AND* ρ ≥ 0**



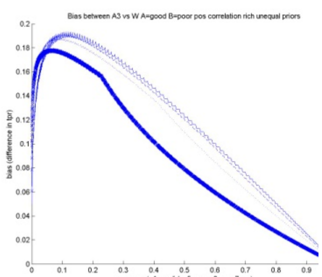

**A=good B=fair *AND* ρ ≥ 0**



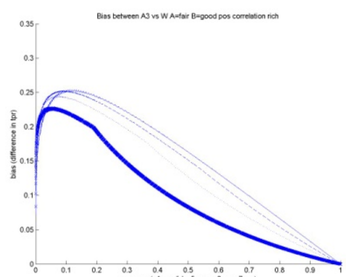

**A=good B=fair *AND* unequal priors ρ ≥ 0**



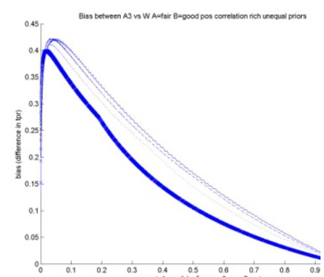

**A=good B=poor *AND* ρ ≥ 0**



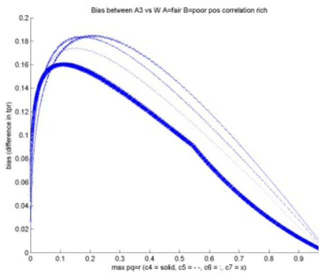

**A=good B=poor *AND* unequal priors ρ ≥ 0**



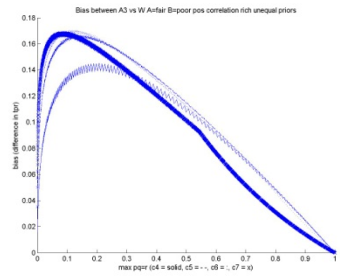

**A=fair B=good *AND* ρ ≥ 0**



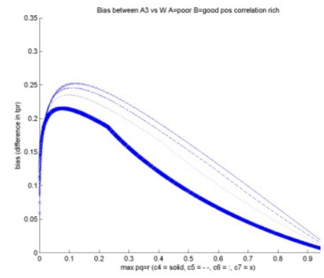

**A=fair B=good *AND* unequal priors ρ ≥ 0**



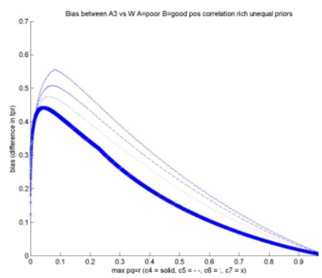

**A=fair B=poor *AND* ρ ≥ 0**



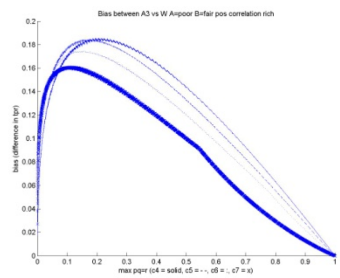

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



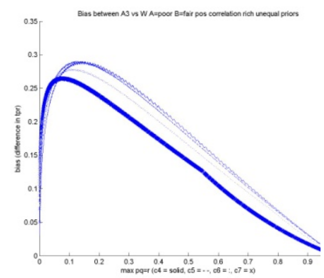

**A=poor B=good *AND* ρ ≥ 0**



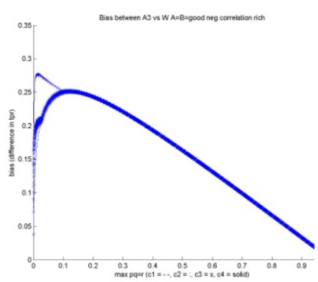

**A=poor B=good *AND* unequal priors ρ ≥ 0**



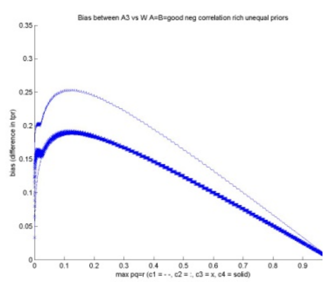

**A=poor B=fair *AND* ρ ≥ 0**



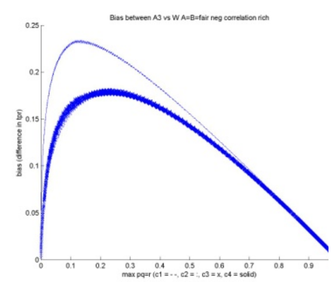

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



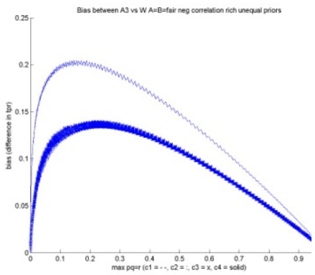

**A=B=good *AND* ρ ≤ 0**



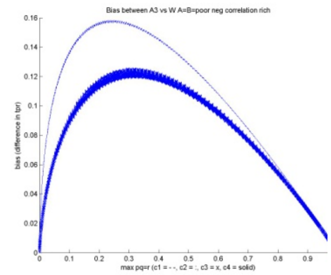

**A=B=good *AND* unequal priors ρ ≤ 0**



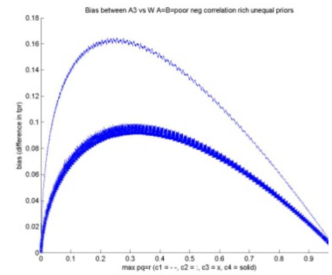

**A=B=fair *AND* ρ ≤ 0**



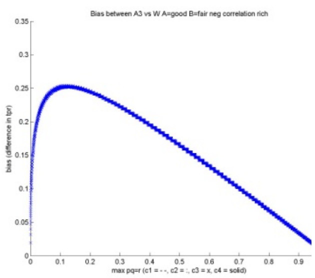

**A=B=fair *AND* unequal priors ρ ≤ 0**



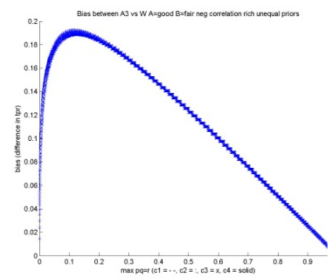

**A=B=poor *AND* ρ ≤ 0**



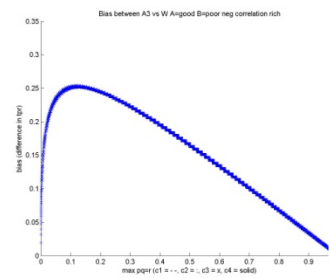

**A=B=poor unequal priors *AND* ρ ≤ 0**



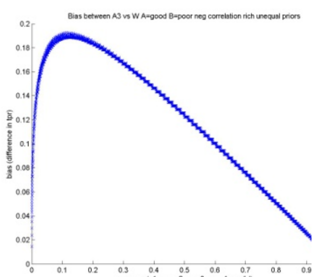

**A=good B=fair *AND* ρ ≤ 0**



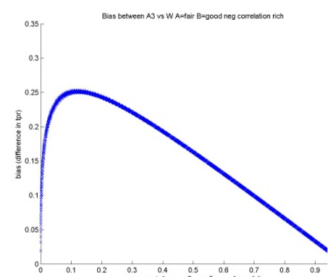

**A=good B=fair *AND* unequal priors ρ ≤ 0**



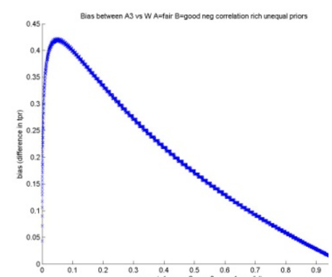

**A=good B=poor *AND* ρ ≤ 0**



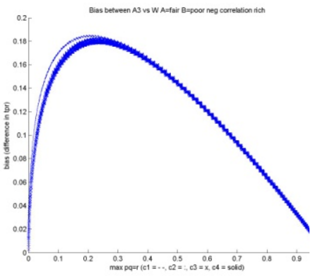

**A=good B=poor *AND* unequal priors ρ ≤ 0**



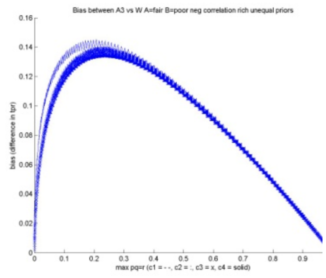

**A=fair B=good *AND* ρ ≤ 0**



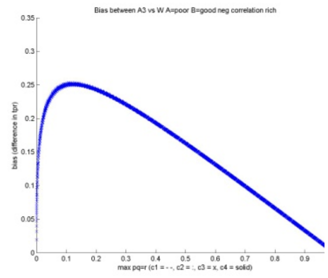

**A=fair B=good *AND* unequal priors ρ ≤ 0**



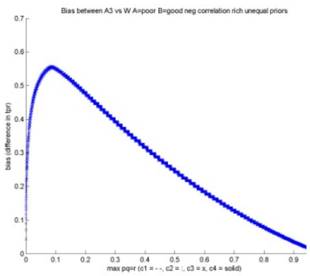

**A=fair B=poor *AND* ρ ≤ 0**



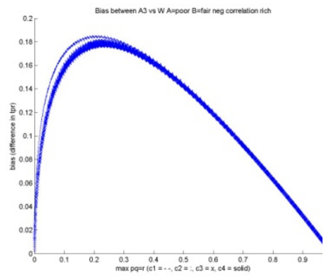

**A=fair B=poor *AND* unequal priors ρ ≤ 0**



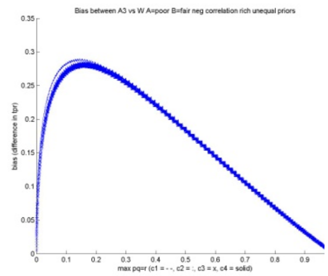

**A=poor B=good *AND* ρ ≤ 0**



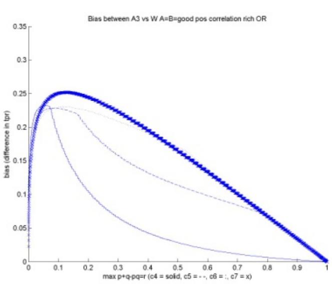

**A=poor B=good *AND* unequal priors ρ ≤ 0**



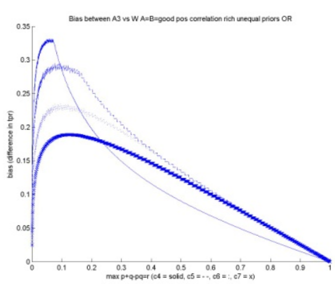

**A=poor B=fair *AND* ρ ≤ 0**



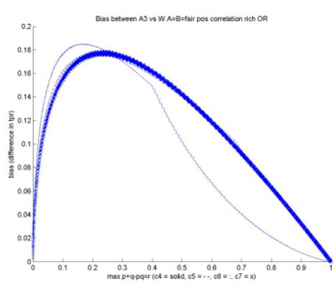

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



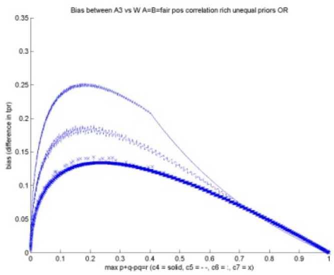

**A=B=good *OR* ρ ≥ 0**



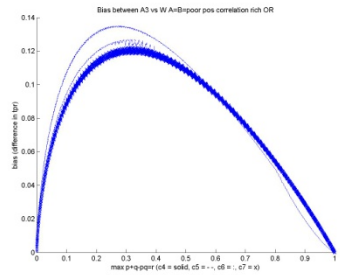

**A=B=good *OR* unequal priors ρ ≥ 0**



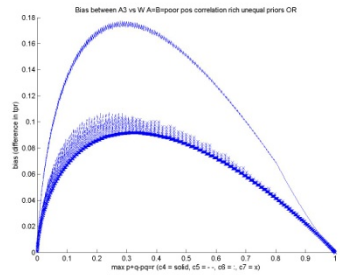

**A=B=fair *OR* ρ ≥ 0**



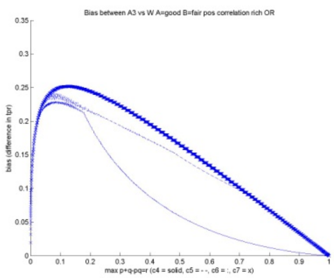

**A=B=fair *OR* unequal priors ρ ≥ 0**



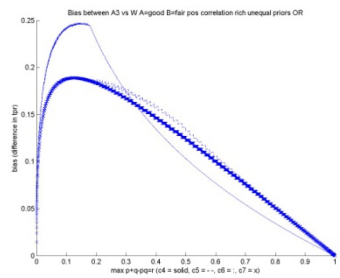

**A=B=poor *OR* ρ ≥ 0**



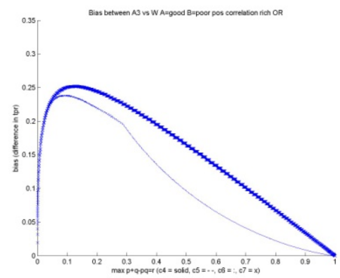

**A=B=poor unequal priors *OR* ρ ≥ 0**



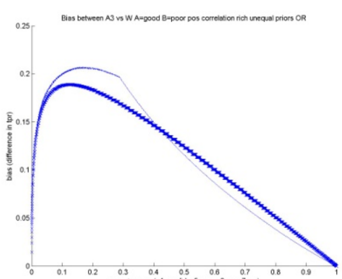

**A=good B=fair *OR* ρ ≥ 0**



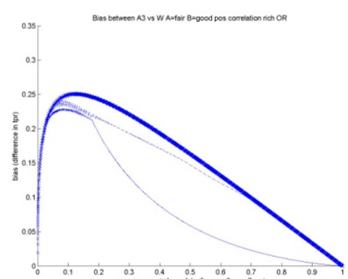

**A=good B=fair *OR* unequal priors ρ ≥ 0**



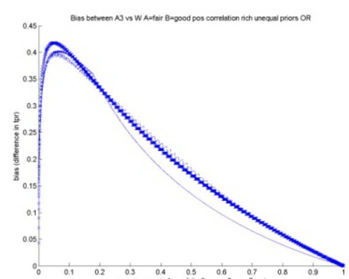

**A=good B=poor *OR* ρ ≥ 0**



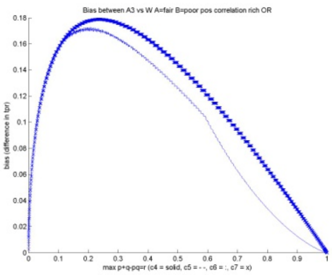

**A=good B=poor *OR* unequal priors ρ ≥ 0**



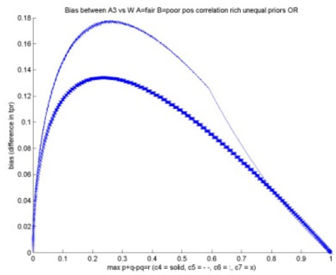

**A=fair B=good *OR* ρ ≥ 0**



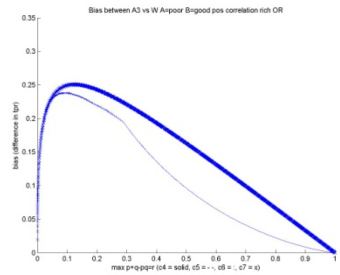

**A=fair B=good *OR* unequal priors ρ ≥ 0**



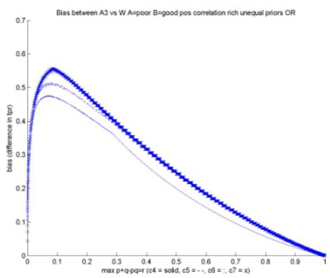

**A=fair B=poor *OR* ρ ≥ 0**



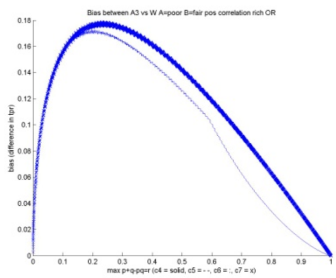

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



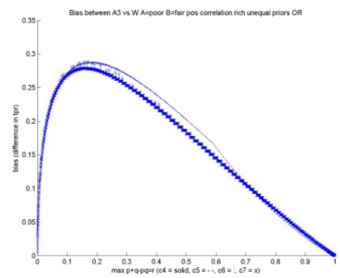

**A=poor B=good *OR* ρ ≥ 0**



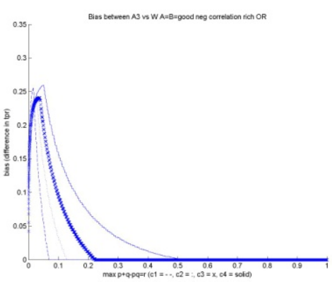

**A=poor B=good *OR* unequal priors ρ ≥ 0**



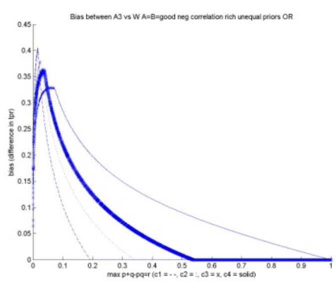

**A=poor B=fair *OR* ρ ≥ 0**



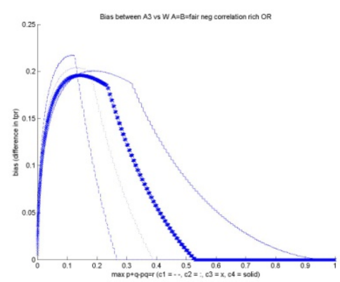

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



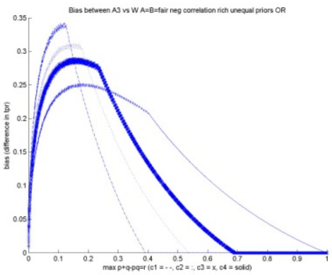

**A=B=good *OR* ρ ≤ 0**



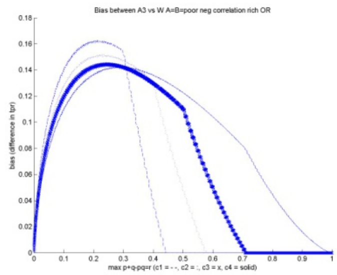

**A=B=good *OR* unequal priors ρ ≤ 0**



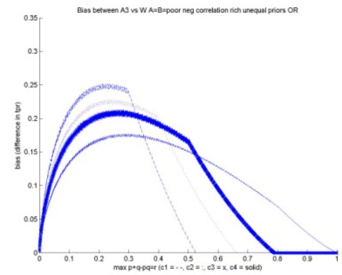

**A=B=fair *OR* ρ ≤ 0**



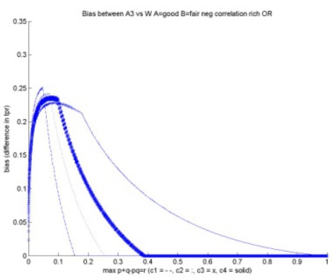
**A=B=fair *OR* unequal priors ρ ≤ 0**



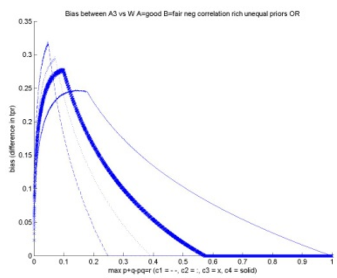
**A=B=poor *OR* ρ ≤ 0**



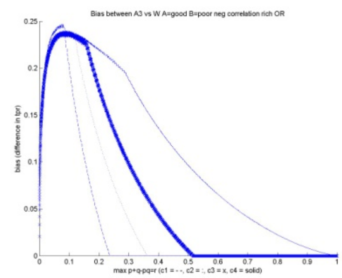
**A=B=poor unequal priors *OR* ρ ≤ 0**



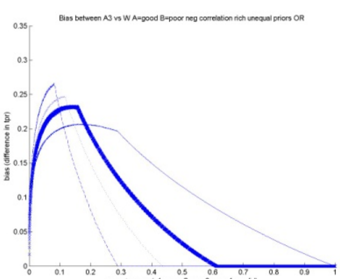
**A=good B=fair *OR* ρ ≤ 0**



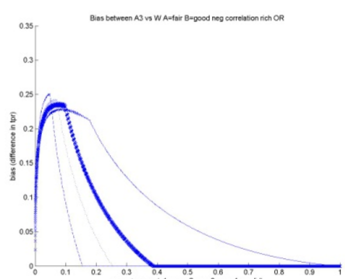
**A=good B=fair *OR* unequal priors ρ ≤ 0**



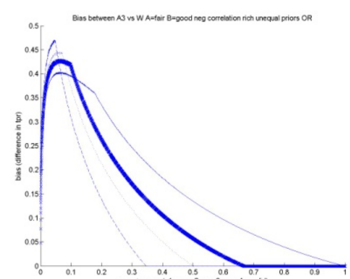
**A=good B=poor *OR* ρ ≤ 0**



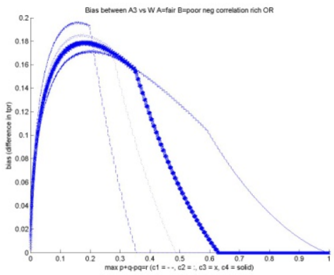
**A=good B=poor *OR* unequal priors ρ ≤ 0**



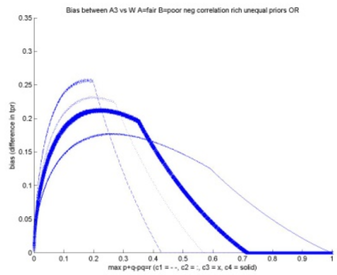
**A=fair B=good *OR* ρ ≤ 0**



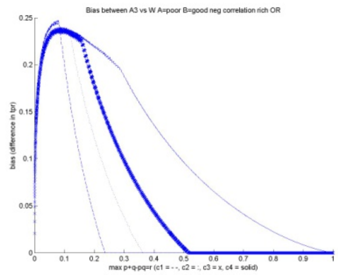
**A=fair B=good *OR* unequal priors ρ ≤ 0**



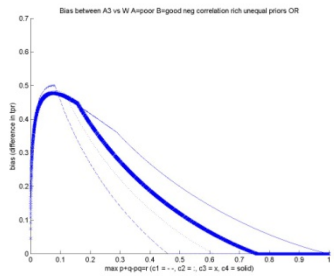
**A=fair B=poor *OR* ρ ≤ 0**



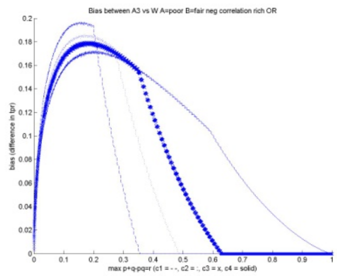
**A=fair B=poor *OR* unequal priors ρ ≤ 0**



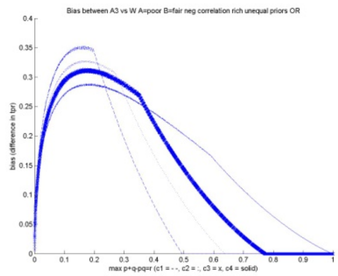
**A=poor B=good *OR* ρ ≤ 0**



**A=poor B=good *OR* unequal priors ρ ≤ 0**



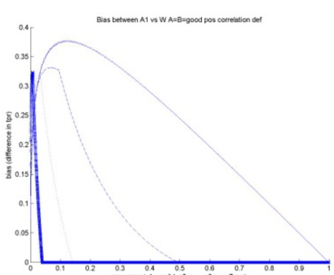
**A=poor B=fair *OR* ρ ≤ 0**



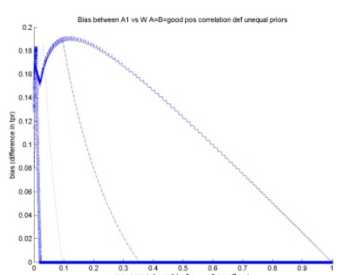
**A=poor B=fair *OR* unequal priors ρ ≤ 0**

## *B.3 – Deficient Target Prevalence*

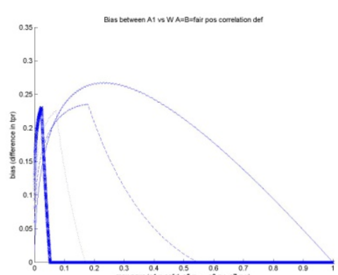
### *B.3.1 – Across I versus Within*



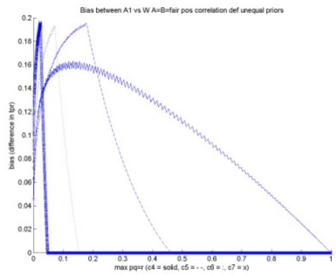
**A=B=good *AND* ρ ≥ 0**



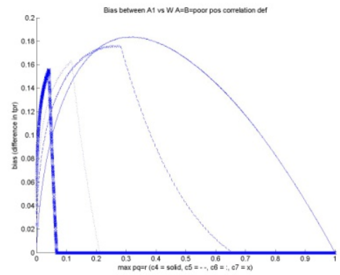
**A=B=good *AND* unequal priors ρ ≥ 0**



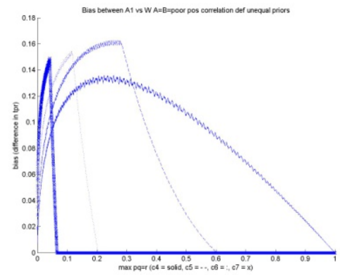
**A=B=fair *AND* ρ ≥ 0**



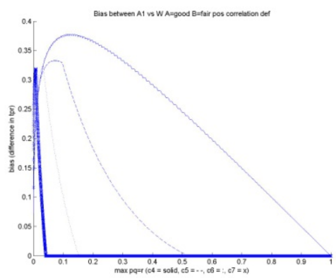

**A=B=fair *AND* unequal priors ρ ≥ 0**



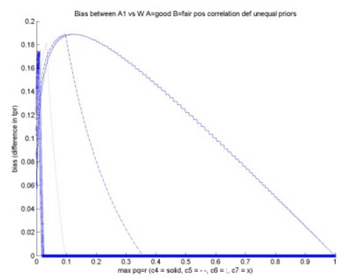

**A=B=poor *AND* ρ ≥ 0**



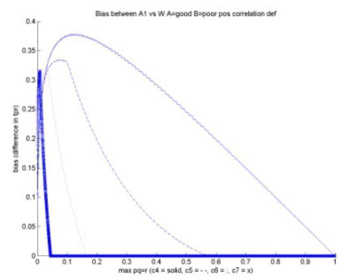

**A=B=poor unequal priors *AND* ρ ≥ 0**



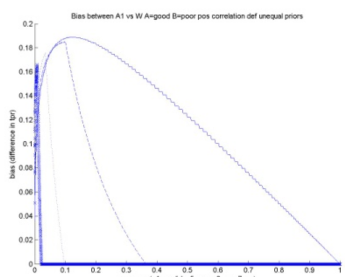

**A=good B=fair *AND* ρ ≥ 0**



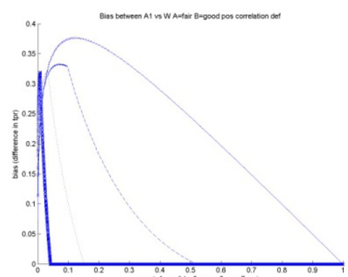

**A=good B=fair *AND* unequal priors ρ ≥ 0**



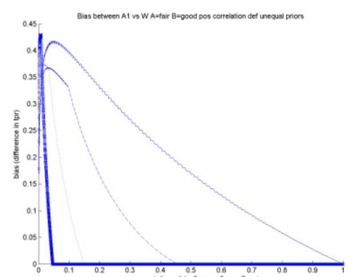

**A=good B=poor *AND* ρ ≥ 0**



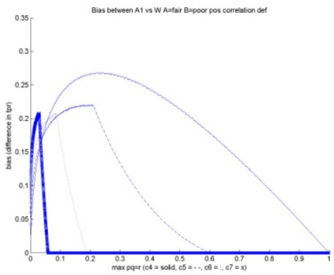

**A=good B=poor *AND* unequal priors ρ ≥ 0**



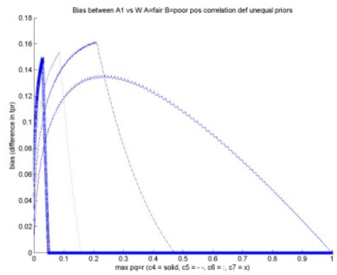

**A=fair B=good *AND* ρ ≥ 0**



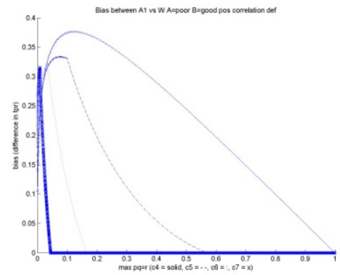

**A=fair B=good *AND* unequal priors ρ ≥ 0**



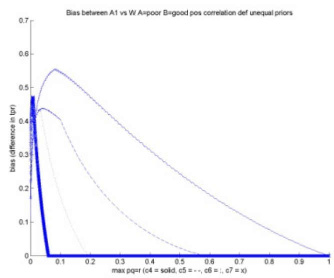

**A=fair B=poor *AND* ρ ≥ 0**



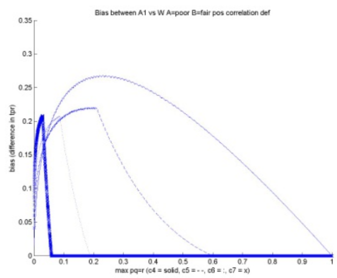

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



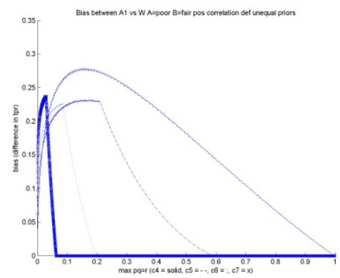

**A=poor B=good *AND* ρ ≥ 0**



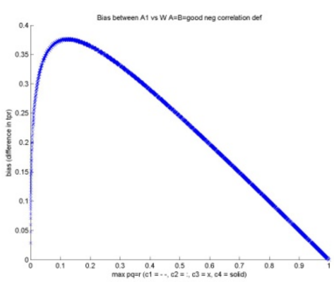

**A=poor B=good *AND* unequal priors ρ ≥ 0**



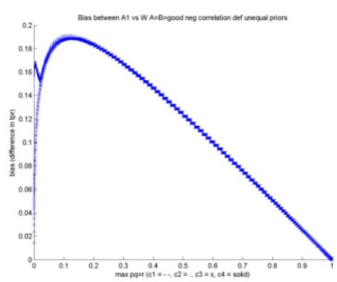

**A=poor B=fair *AND* ρ ≥ 0**



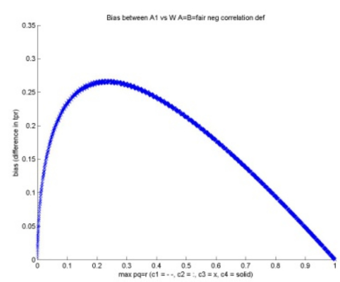

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



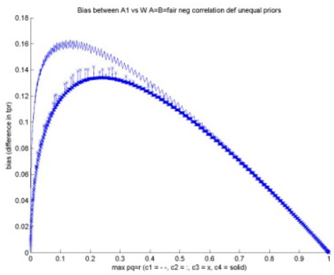

**A=B=good *AND* ρ ≤ 0**



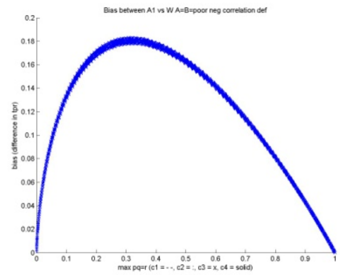

**A=B=good *AND* unequal priors ρ ≤ 0**



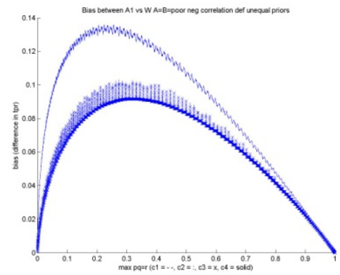

**A=B=fair *AND* ρ ≤ 0**



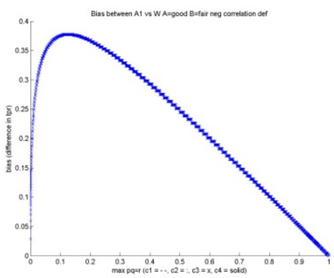
**A=B=fair *AND* unequal priors ρ ≤ 0**



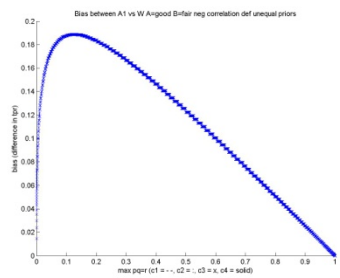
**A=B=poor *AND* ρ ≤ 0**



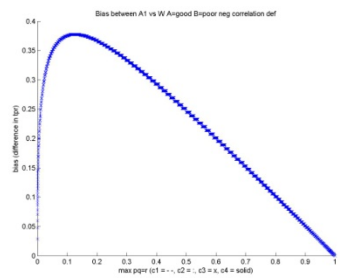
**A=B=poor unequal priors *AND* ρ ≤ 0**



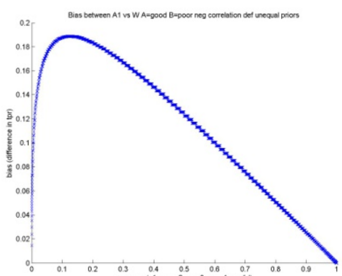
**A=good B=fair *AND* ρ ≤ 0**



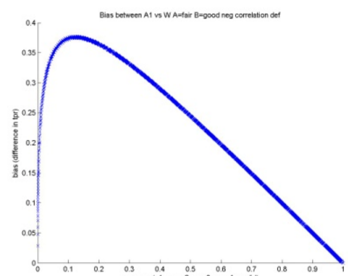
**A=good B=fair *AND* unequal priors ρ ≤ 0**



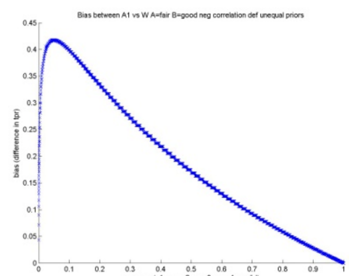
**A=good B=poor *AND* ρ ≤ 0**



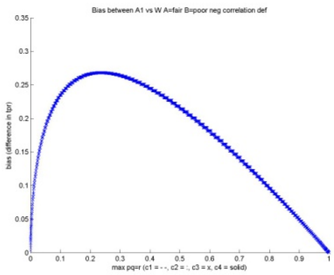
**A=good B=poor *AND* unequal priors ρ ≤ 0**



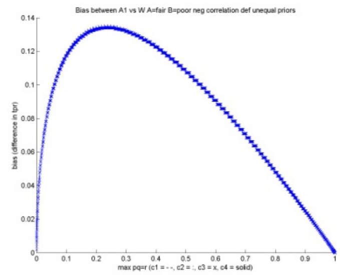
**A=fair B=good *AND* ρ ≤ 0**



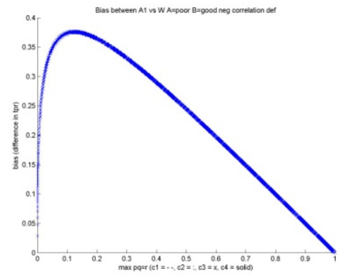
**A=fair B=good *AND* unequal priors ρ ≤ 0**



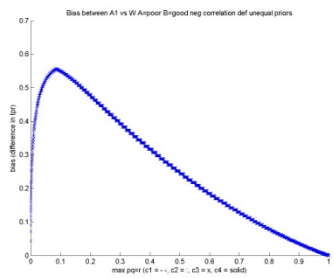

**A=fair B=poor *AND* ρ ≤ 0**



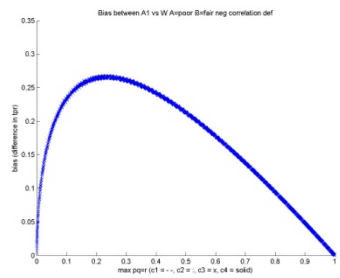

**A=fair B=poor *AND* unequal priors ρ ≤ 0**



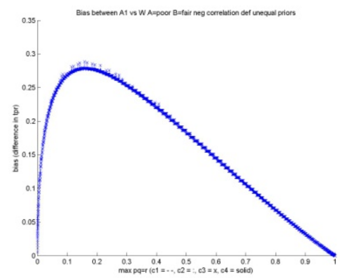

**A=poor B=good *AND* ρ ≤ 0**



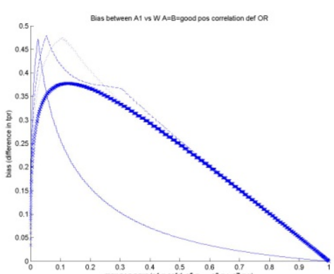

**A=poor B=good *AND* unequal priors ρ ≤ 0**



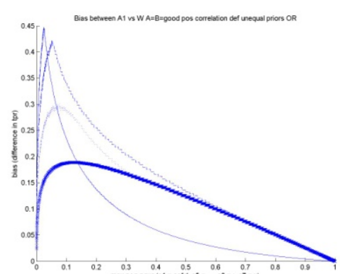

**A=poor B=fair *AND* ρ ≤ 0**



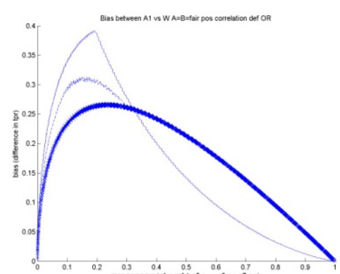

**A=poor B=fair *AND* unequal priors ρ ≤ 0**



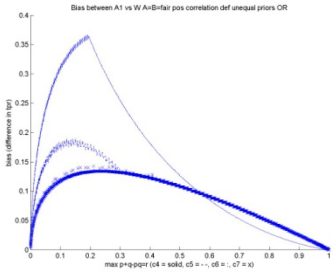

**A=B=good *OR* ρ ≥ 0**



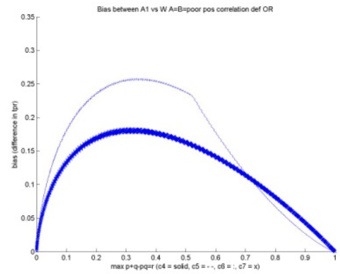

**A=B=good *OR* unequal priors ρ ≥ 0**



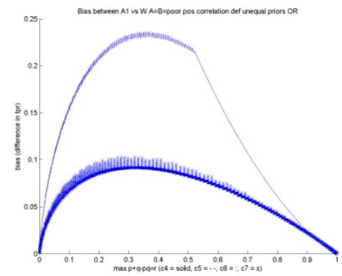

**A=B=fair *OR* ρ ≥ 0**



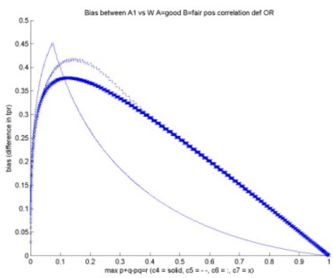
**A=B=fair *OR* unequal priors ρ ≥ 0**



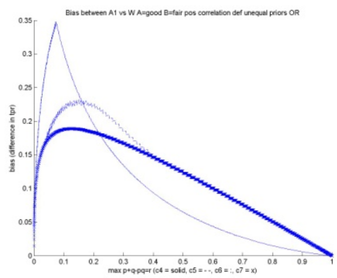
**A=B=poor *OR* ρ ≥ 0**



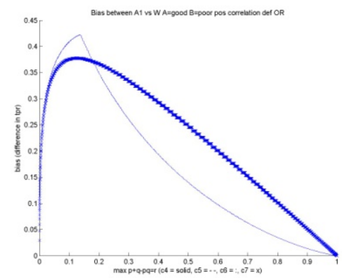
**A=B=poor unequal priors *OR* ρ ≥ 0**



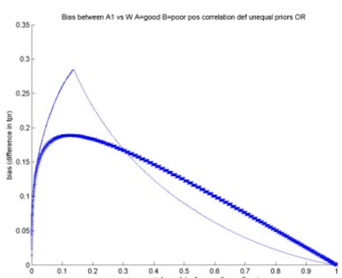
**A=good B=fair *OR* ρ ≥ 0**



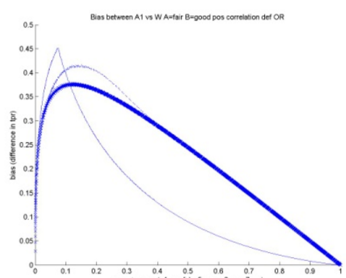
**A=good B=fair *OR* unequal priors ρ ≥ 0**



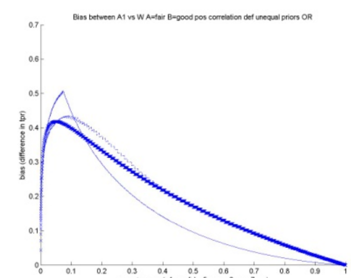
**A=good B=poor *OR* ρ ≥ 0**



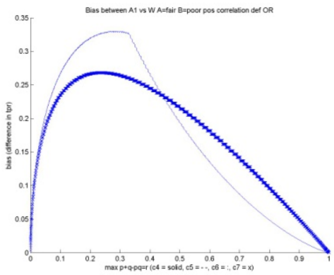
**A=good B=poor *OR* unequal priors ρ ≥ 0**



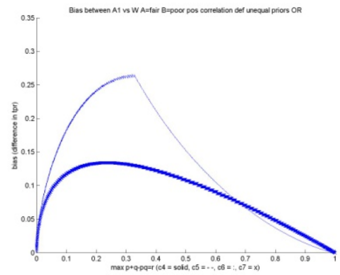
**A=fair B=good *OR* ρ ≥ 0**



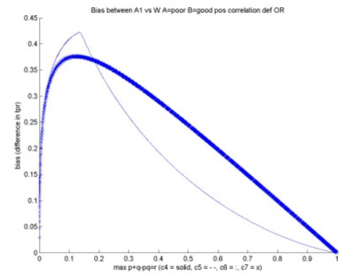
**A=fair B=good *OR* unequal priors ρ ≥ 0**



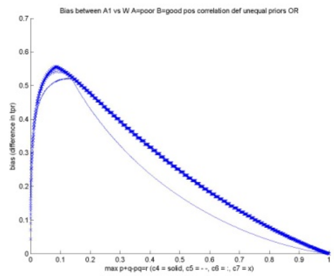

**A=fair B=poor *OR* ρ ≥ 0**



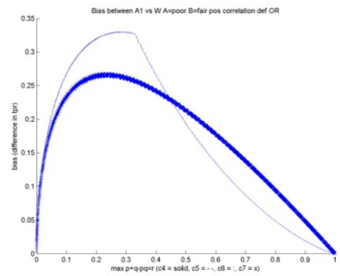

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



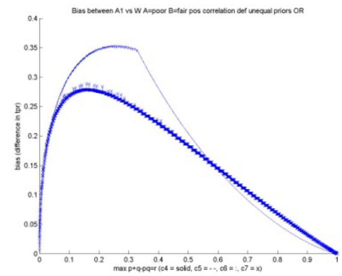

**A=poor B=good *OR* ρ ≥ 0**



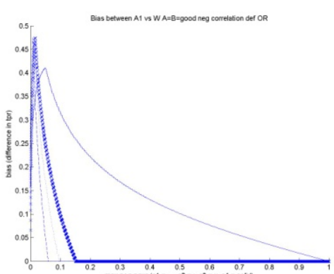

**A=poor B=good *OR* unequal priors ρ ≥ 0**



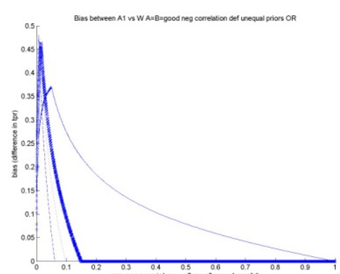

**A=poor B=fair *OR* ρ ≥ 0**



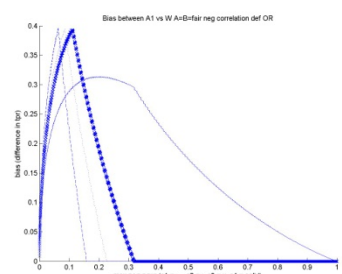

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



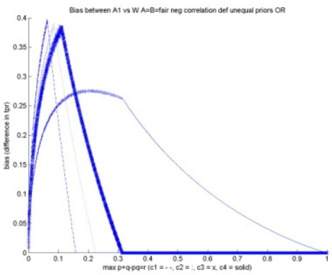

**A=B=good *OR* ρ ≤ 0**



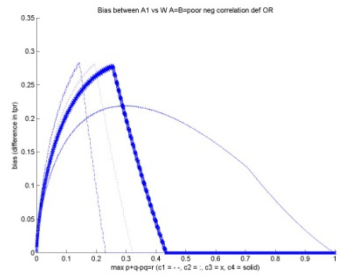

**A=B=good *OR* unequal priors ρ ≤ 0**



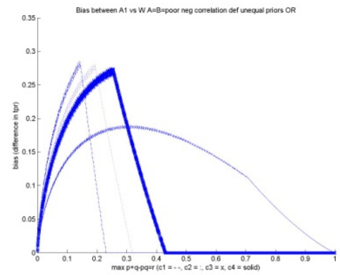

**A=B=fair *OR* ρ ≤ 0**



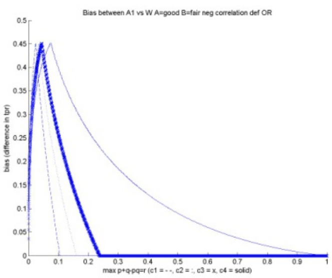

**A=B=fair *OR* unequal priors ρ ≤ 0**



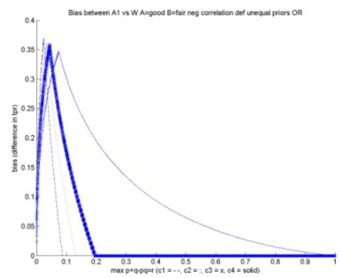

**A=B=poor *OR* ρ ≤ 0**



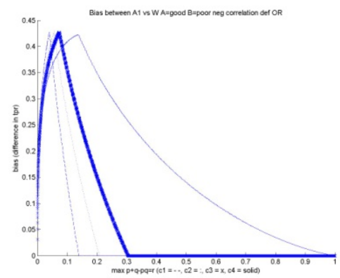

**A=B=poor unequal priors *OR* ρ ≤ 0**



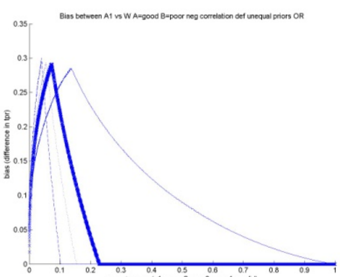

**A=good B=fair *OR* ρ ≤ 0**



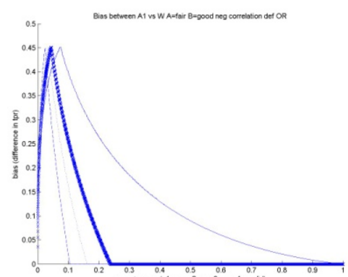

**A=good B=fair *OR* unequal priors ρ ≤ 0**



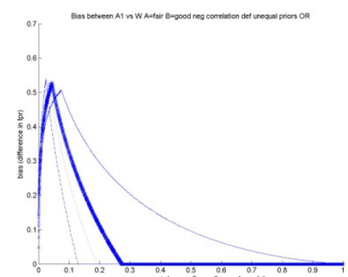

**A=good B=poor *OR* ρ ≤ 0**



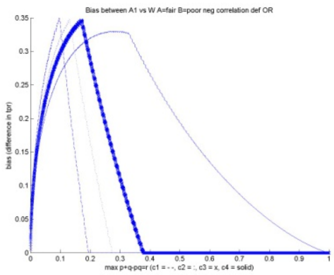

**A=good B=poor *OR* unequal priors ρ ≤ 0**



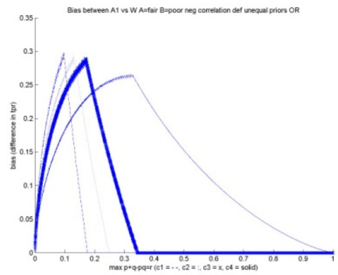

**A=fair B=good *OR* ρ ≤ 0**



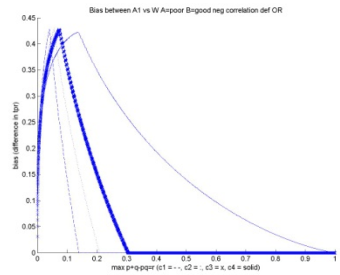

**A=fair B=good *OR* unequal priors ρ ≤ 0**



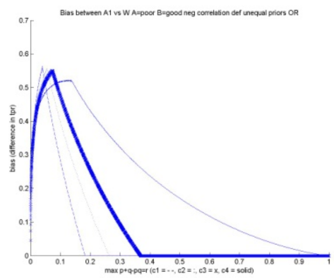

**A=fair B=poor *OR* ρ ≤ 0**



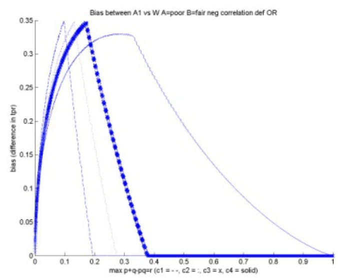

**A=fair B=poor *OR* unequal priors ρ ≤ 0**



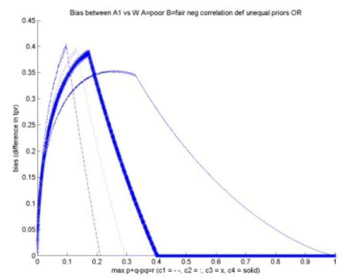

**A=poor B=good *OR* ρ ≤ 0**




**A=poor B=good *OR* unequal priors ρ ≤ 0**

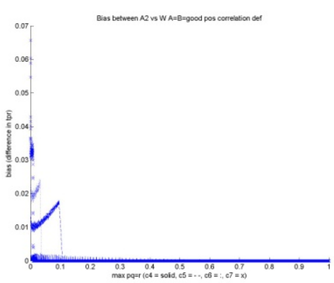



**A=poor B=fair *OR* ρ ≤ 0**

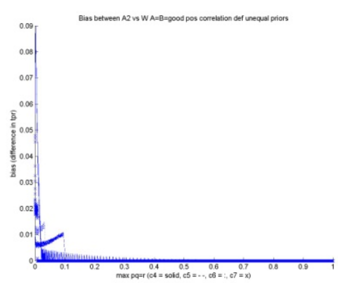



**A=poor B=fair *OR* unequal priors ρ ≤ 0**

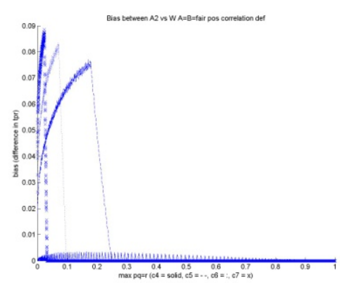
### *B.3.2 – Across II versus Within*



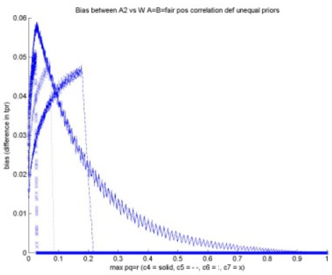

**A=B=good *AND* ρ ≥ 0**



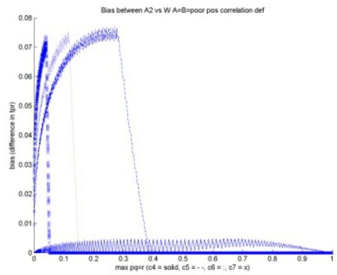

**A=B=good *AND* unequal priors ρ ≥ 0**



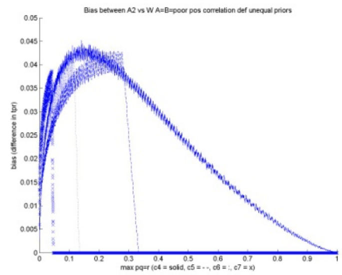

**A=B=fair *AND* ρ ≥ 0**



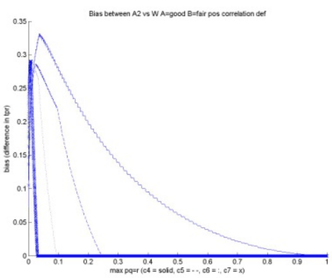

**A=B=fair *AND* unequal priors ρ ≥ 0**



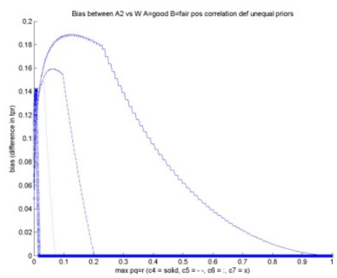

**A=B=poor *AND* ρ ≥ 0**



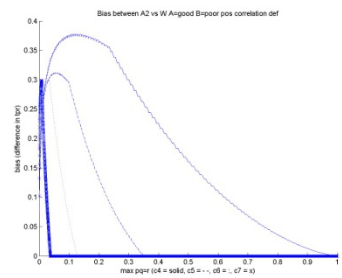

**A=B=poor unequal priors *AND* ρ ≥ 0**



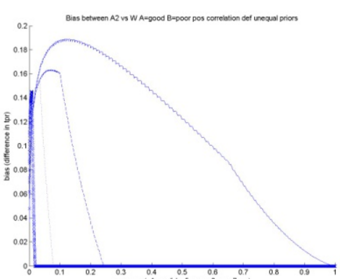

**A=good B=fair *AND* ρ ≥ 0**



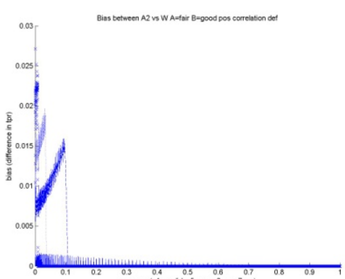

**A=good B=fair *AND* unequal priors ρ ≥ 0**



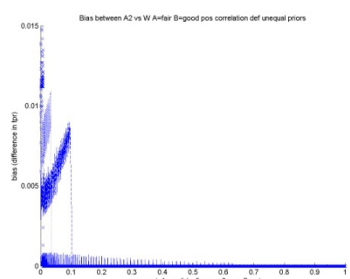

**A=good B=poor *AND* ρ ≥ 0**



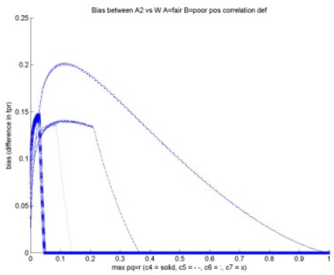

**A=good B=poor *AND* unequal priors ρ ≥ 0**

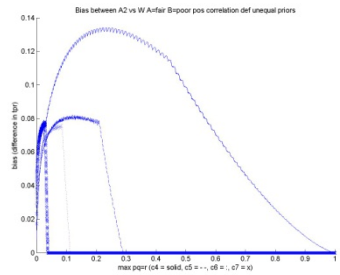


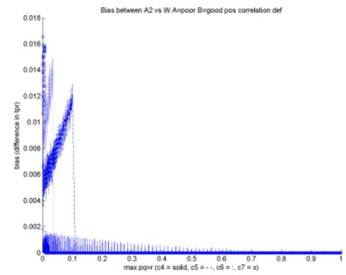

**A=fair B=good *AND* ρ ≥ 0**




**A=fair B=good *AND* unequal priors ρ ≥ 0**



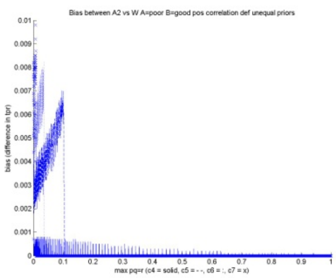



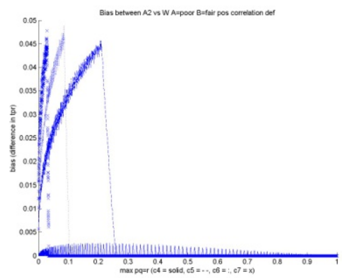



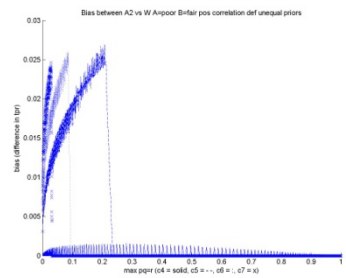

**A=fair B=poor *AND* ρ ≥ 0**

**A=fair B=poor *AND* unequal priors ρ ≥ 0**

**A=poor B=good *AND* ρ ≥ 0**



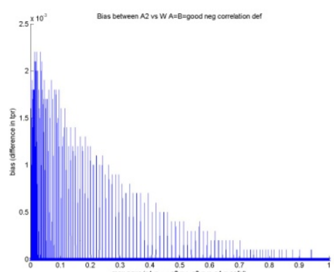



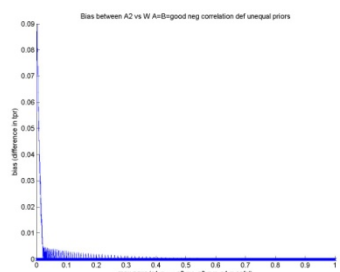



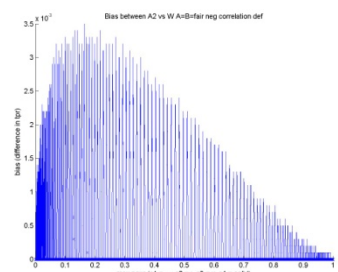

**A=poor B=good *AND* unequal priors ρ ≥ 0**

**A=poor B=fair *AND* ρ ≥ 0**

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



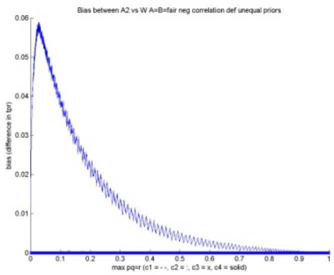



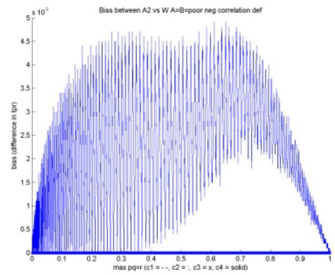

**A=B=good *AND* ρ ≤ 0**

**A=B=good *AND* unequal priors ρ ≤ 0**

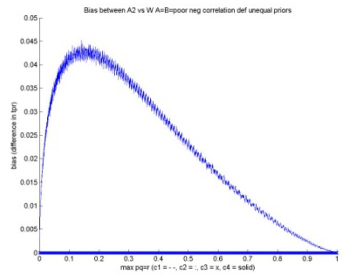
**A=B=fair *AND* ρ ≤ 0**



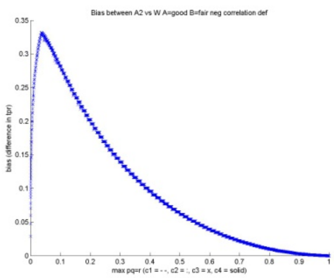

**A=B=fair *AND* unequal priors ρ ≤ 0**



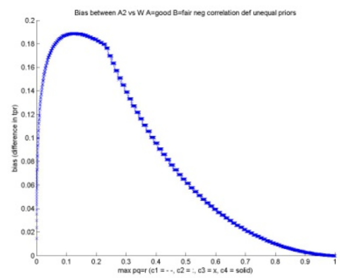

**A=B=poor *AND* ρ ≤ 0**



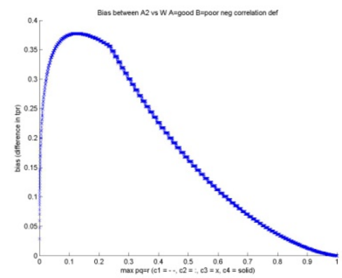

**A=B=poor unequal priors *AND* ρ ≤ 0**



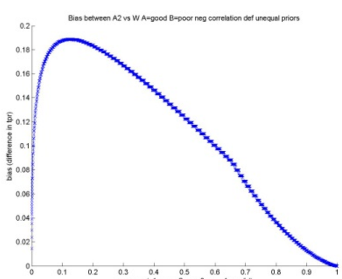

**A=good B=fair *AND* ρ ≤ 0**



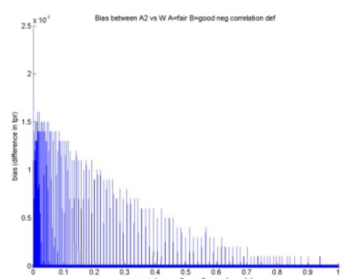

**A=good B=fair *AND* unequal priors ρ ≤ 0**



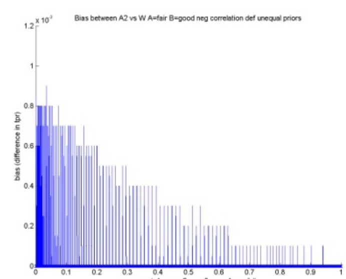

**A=good B=poor *AND* ρ ≤ 0**



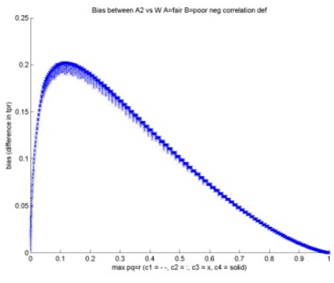

**A=good B=poor *AND* unequal priors ρ ≤ 0**



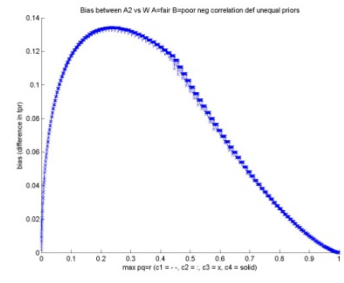

**A=fair B=good *AND* ρ ≤ 0**



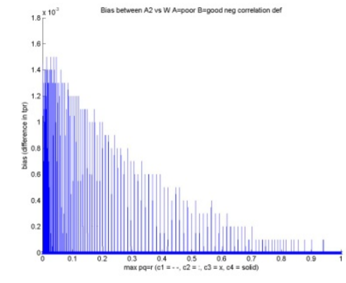

**A=fair B=good *AND* unequal priors ρ ≤ 0**



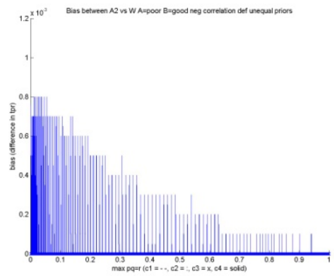
**A=fair B=poor *AND* ρ ≤ 0**



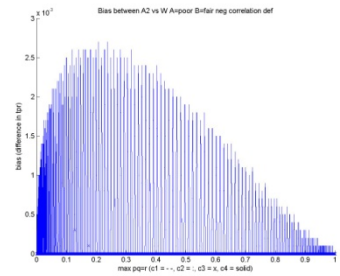
**A=fair B=poor *AND* unequal priors ρ ≤ 0**



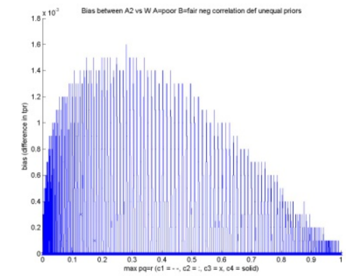
**A=poor B=good *AND* ρ ≤ 0**



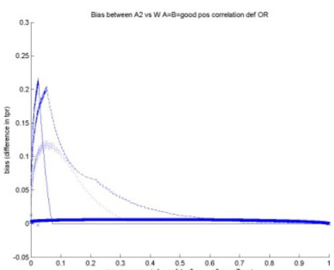
**A=poor B=good *AND* unequal priors ρ ≤ 0**



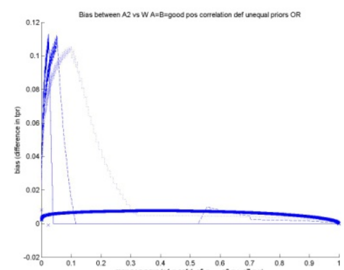
**A=poor B=fair *AND* ρ ≤ 0**



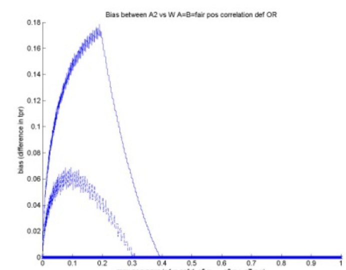
**A=poor B=fair *AND* unequal priors ρ ≤ 0**



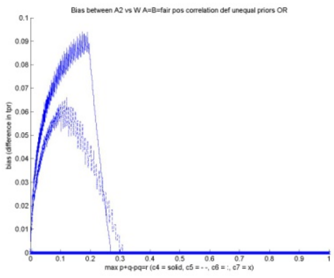
**(All values > 0) A=B=good *OR* ρ ≥ 0**



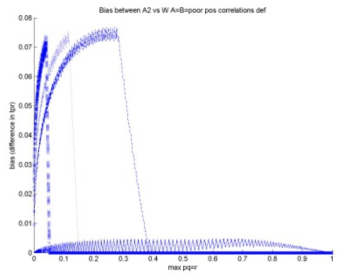
**(All values > 0) A=B=good *OR* unequal priors ρ ≥ 0**



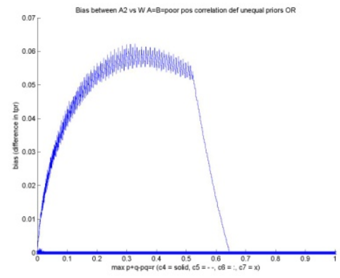
**A=B=fair *OR* ρ ≥ 0**



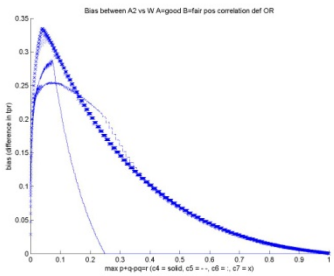

**A=B=fair *OR* unequal priors ρ ≥ 0**



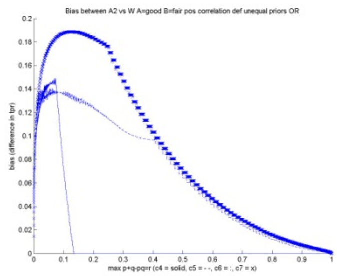

**A=B=poor *OR* ρ ≥ 0**



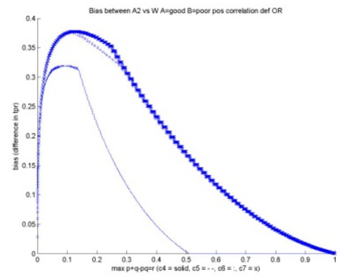

**A=B=poor unequal priors *OR* ρ ≥ 0**



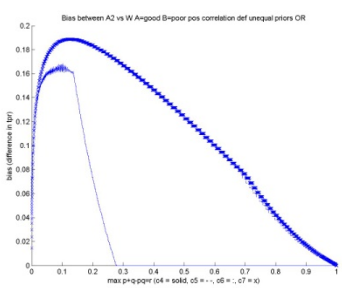

**A=good B=fair *OR* ρ ≥ 0**



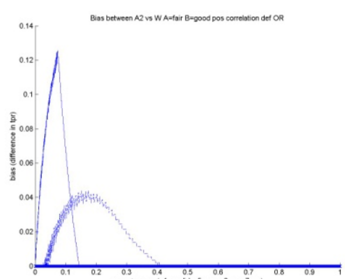

**A=good B=fair *OR* unequal priors ρ ≥ 0**



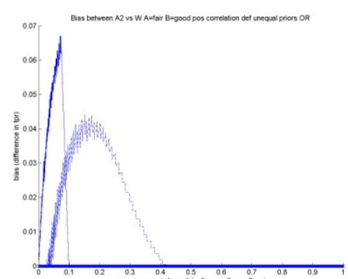

**A=good B=poor *OR* ρ ≥ 0**



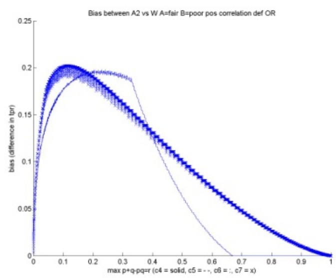

**A=good B=poor *OR* unequal priors ρ ≥ 0**



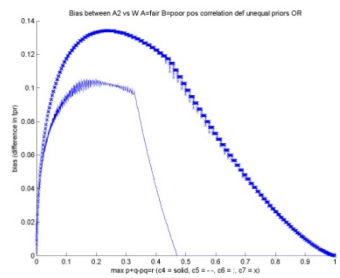

**A=fair B=good *OR* ρ ≥ 0**



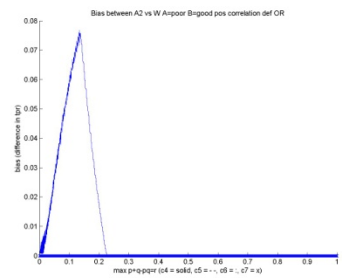

**A=fair B=good *OR* unequal priors ρ ≥ 0**



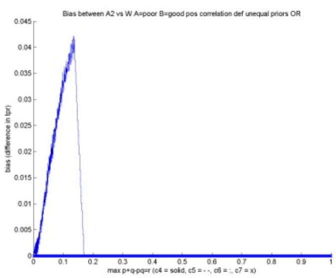

**A=fair B=poor *OR* ρ ≥ 0**



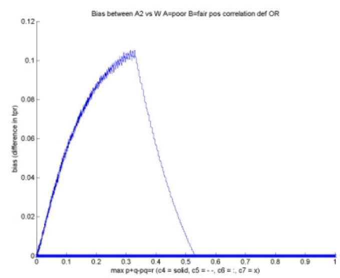

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



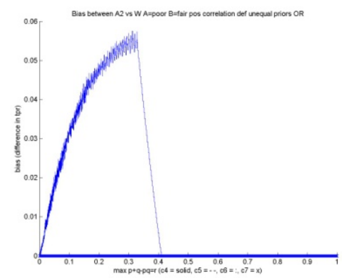

**A=poor B=good *OR* ρ ≥ 0**



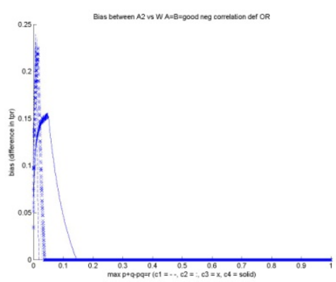

**A=poor B=good *OR* unequal priors ρ ≥ 0**



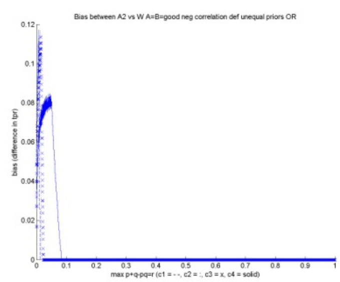

**A=poor B=fair *OR* ρ ≥ 0**



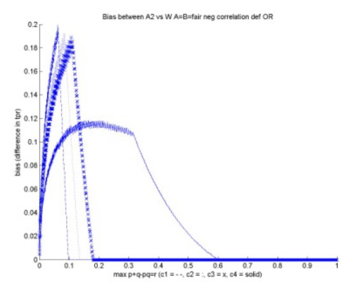

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



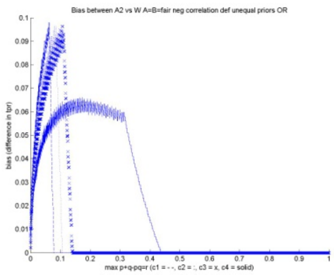

**A=B=good *OR* ρ ≤ 0**



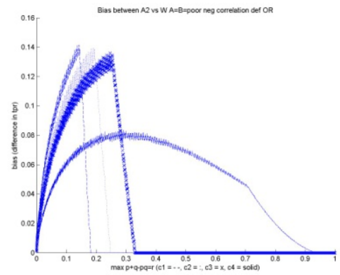

**A=B=good *OR* unequal priors ρ ≤ 0**



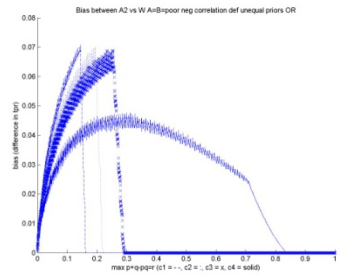

**A=B=fair *OR* ρ ≤ 0**



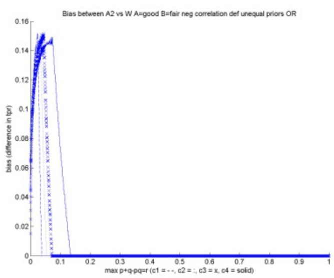

**A=B=fair *OR* unequal priors ρ ≤ 0**



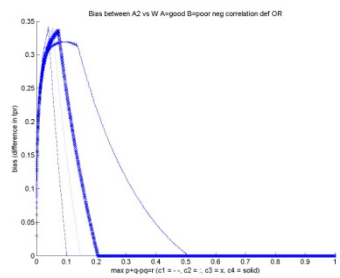

**A=B=poor *OR* ρ ≤ 0**



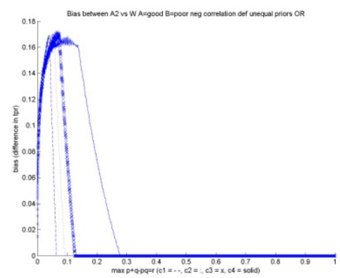

**A=B=poor unequal priors *OR* ρ ≤ 0**



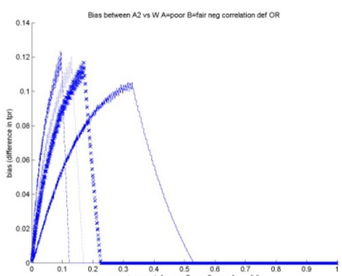

**A=good B=fair *OR* ρ ≤ 0**



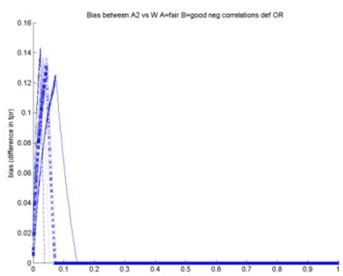

**A=good B=fair *OR* unequal priors ρ ≤ 0**



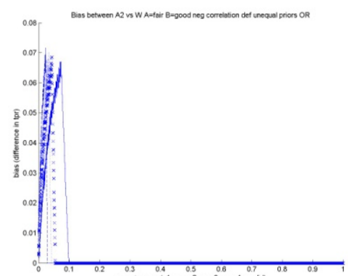

**A=good B=poor *OR* ρ ≤ 0**



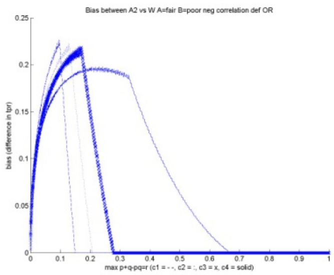

**A=good B=poor *OR* unequal priors ρ ≤ 0**



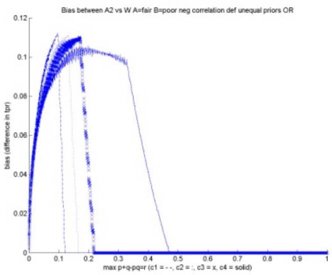

**A=fair B=good *OR* ρ ≤ 0**



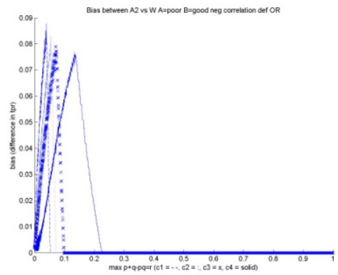

**A=fair B=good *OR* unequal priors ρ ≤ 0**



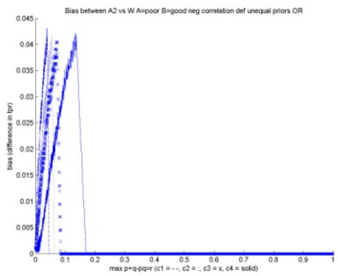

**A=fair B=poor *OR* ρ ≤ 0**



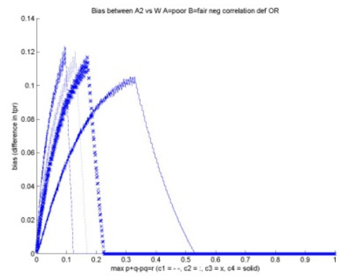

**A=fair B=poor *OR* unequal priors ρ ≤ 0**



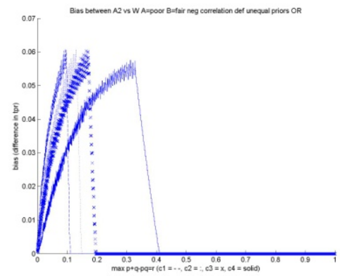

**A=poor B=good *OR* ρ ≤ 0**




**A=poor B=good *OR* unequal priors ρ ≤ 0**

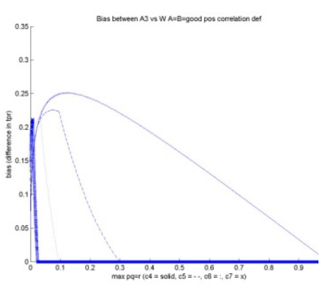



**A=poor B=fair *OR* ρ ≤ 0**

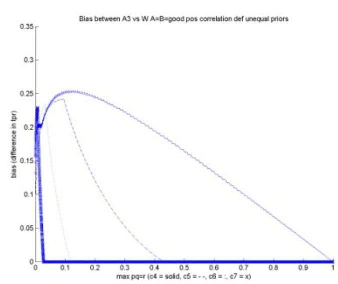



**A=poor B=fair *OR* unequal priors ρ ≤ 0**

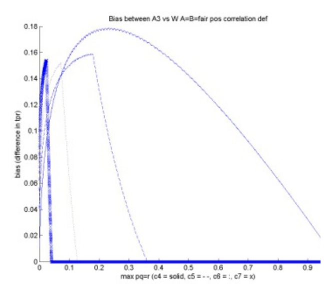
### *B.3.3 – Across III versus Within*



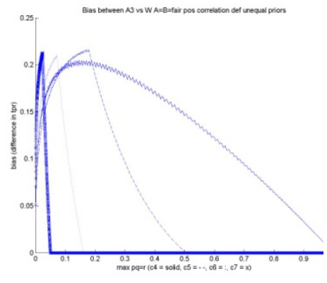

**A=B=good *AND* ρ ≥ 0**



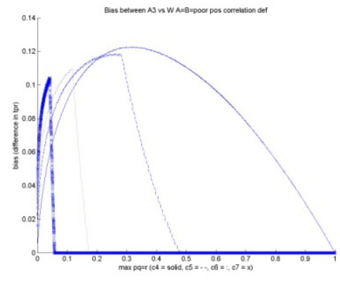

**A=B=good *AND* unequal priors ρ ≥ 0**



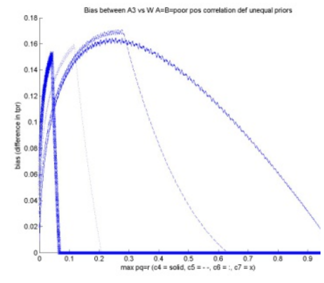

**A=B=fair *AND* ρ ≥ 0**



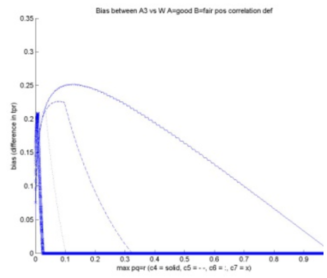

**A=B=fair *AND* unequal priors ρ ≥ 0**



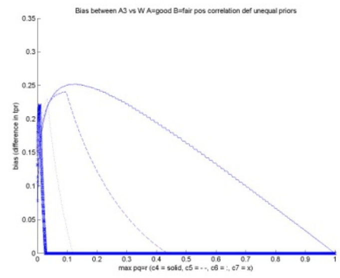

**A=B=poor *AND* ρ ≥ 0**



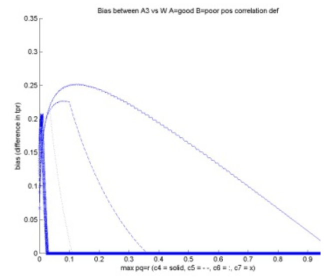

**A=B=poor unequal priors *AND* ρ ≥ 0**



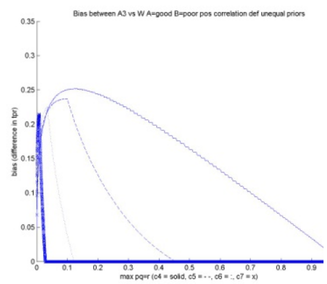

**A=good B=fair *AND* ρ ≥ 0**



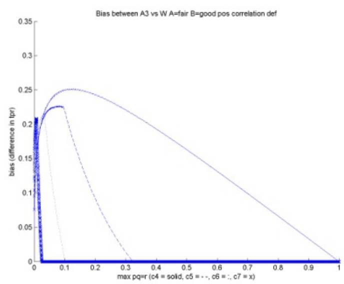

**A=good B=fair *AND* unequal priors ρ ≥ 0**



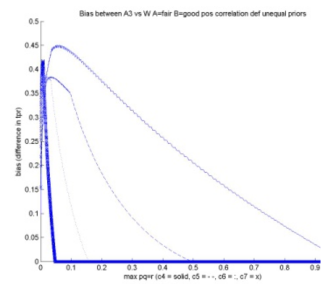

**A=good B=poor *AND* ρ ≥ 0**



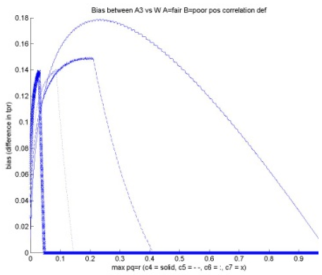

**A=good B=poor *AND* unequal priors ρ ≥ 0**



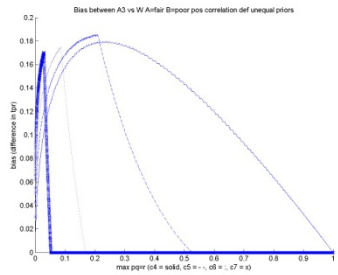

**A=fair B=good *AND* ρ ≥ 0**



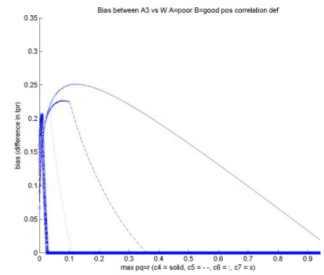

**A=fair B=good *AND* unequal priors ρ ≥ 0**



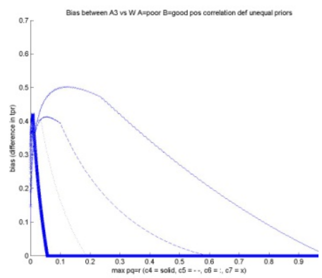

**A=fair B=poor *AND* ρ ≥ 0**



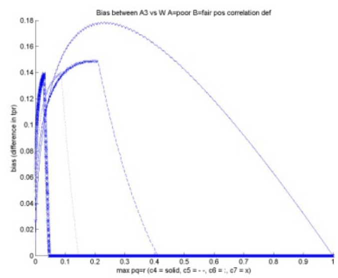

**A=fair B=poor *AND* unequal priors ρ ≥ 0**



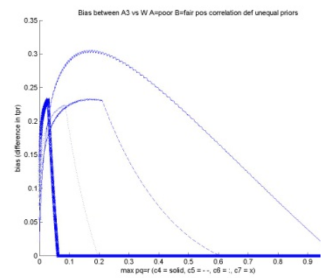

**A=poor B=good *AND* ρ ≥ 0**



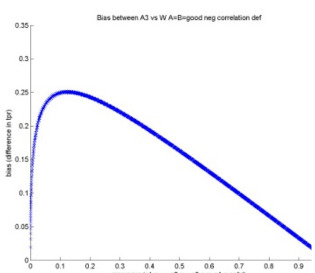

**A=poor B=good *AND* unequal priors ρ ≥ 0**



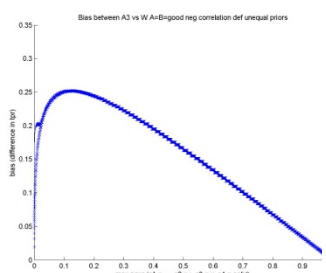

**A=poor B=fair *AND* ρ ≥ 0**



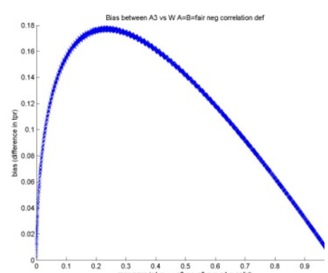

**A=poor B=fair *AND* unequal priors ρ ≥ 0**



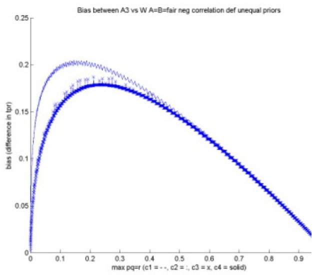

**A=B=good *AND* ρ ≤ 0**



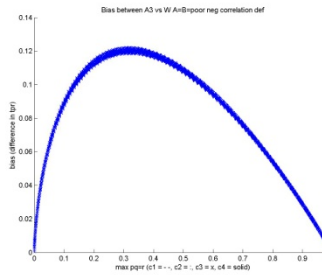

**A=B=good *AND* unequal priors ρ ≤ 0**



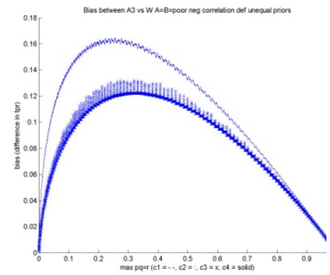

**A=B=fair *AND* ρ ≤ 0**



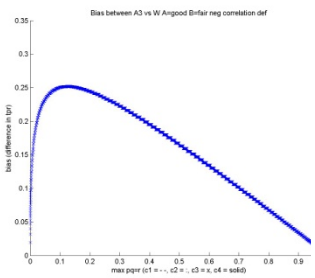

**A=B=fair *AND* unequal priors ρ ≤ 0**



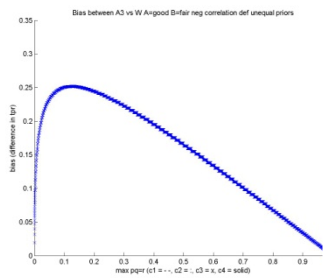

**A=B=poor *AND* ρ ≤ 0**



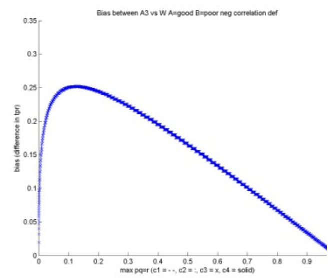

**A=B=poor unequal priors *AND* ρ ≤ 0**



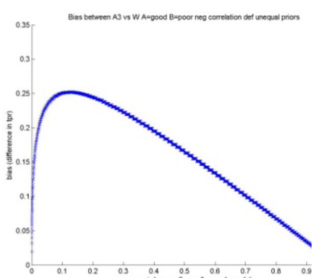

**A=good B=fair *AND* ρ ≤ 0**



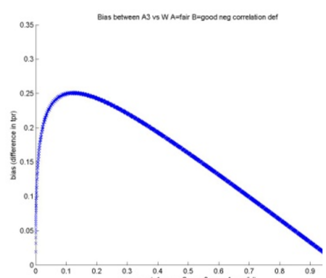

**A=good B=fair *AND* unequal priors ρ ≤ 0**



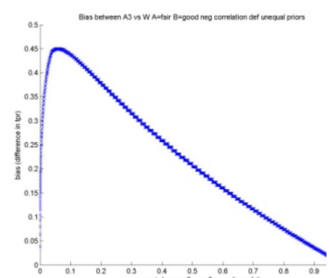

**A=good B=poor *AND* ρ ≤ 0**



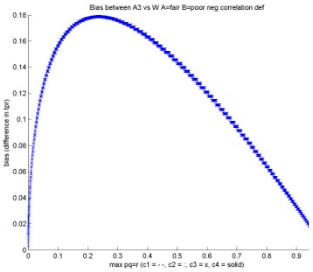

**A=good B=poor *AND* unequal priors ρ ≤ 0**



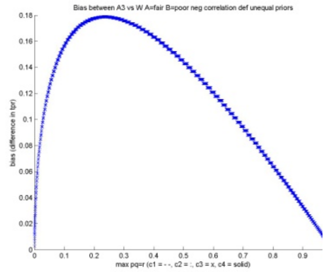

**A=fair B=good *AND* ρ ≤ 0**



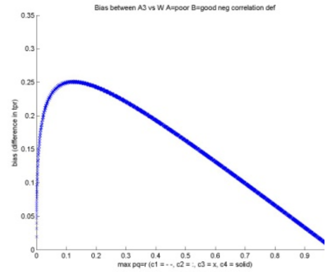

**A=fair B=good *AND* unequal priors ρ ≤ 0**



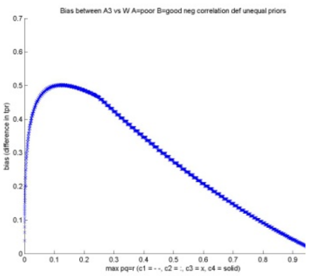
**A=fair B=poor *AND* ρ ≤ 0**



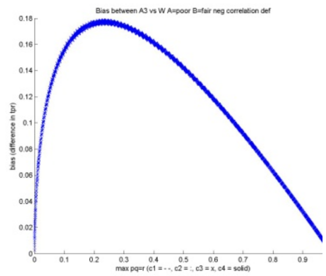
**A=fair B=poor *AND* unequal priors ρ ≤ 0**



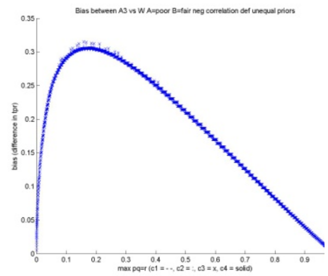
**A=poor B=good *AND* ρ ≤ 0**



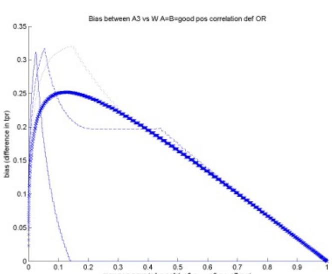
**A=poor B=good *AND* unequal priors ρ ≤ 0**



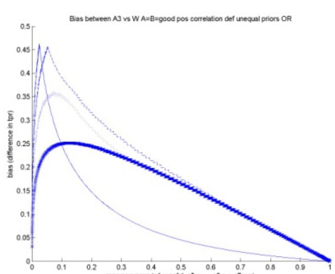
**A=poor B=fair *AND* ρ ≤ 0**



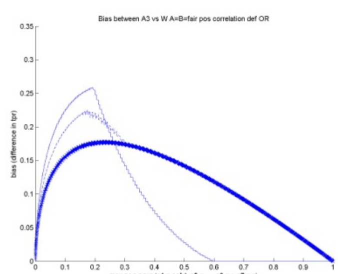
**A=poor B=fair *AND* unequal priors ρ ≤ 0**



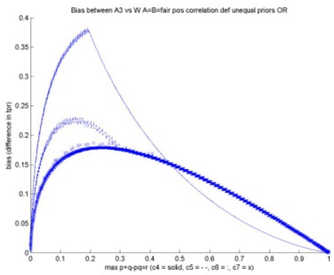
**A=B=good *OR* ρ ≥ 0**



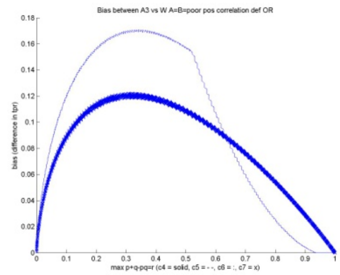
**A=B=good *OR* unequal priors ρ ≥ 0**



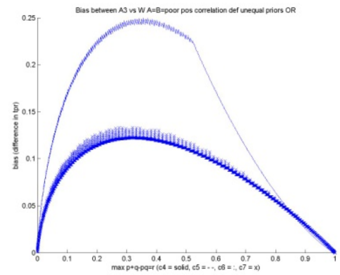
**A=B=fair *OR* ρ ≥ 0**



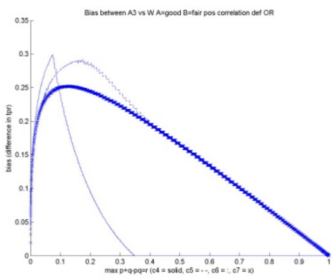

**A=B=fair *OR* unequal priors ρ ≥ 0**



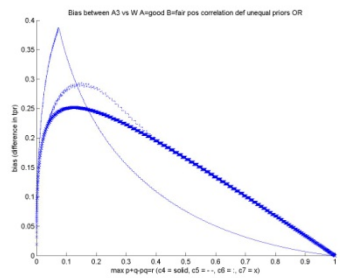

**A=B=poor *OR* ρ ≥ 0**



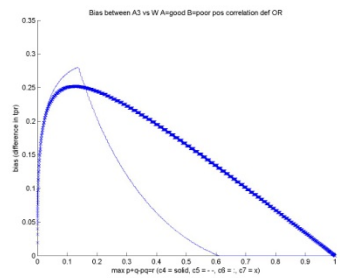

**A=B=poor unequal priors *OR* ρ ≥ 0**



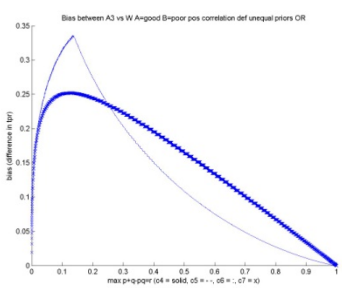

**A=good B=fair *OR* ρ ≥ 0**



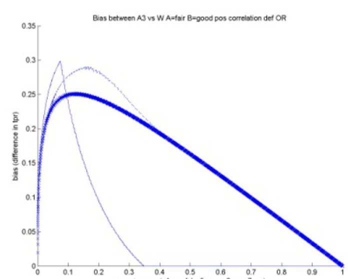

**A=good B=fair *OR* unequal priors ρ ≥ 0**



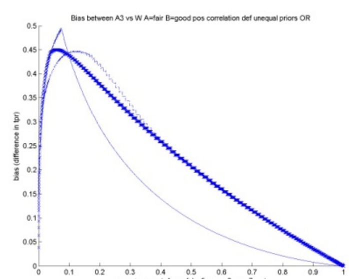

**A=good B=poor *OR* ρ ≥ 0**



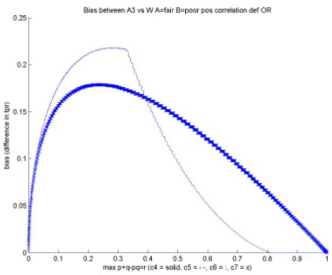

**A=good B=poor *OR* unequal priors ρ ≥ 0**



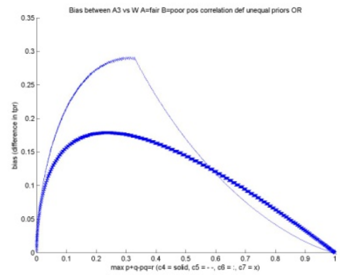

**A=fair B=good *OR* ρ ≥ 0**



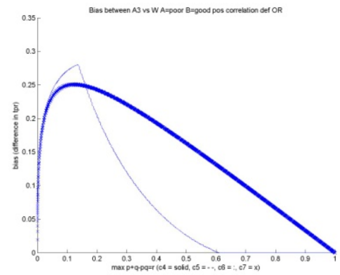

**A=fair B=good *OR* unequal priors ρ ≥ 0**



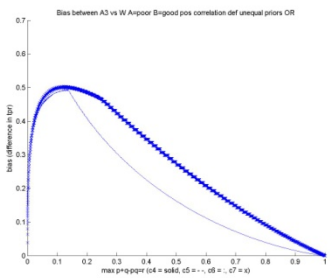

**A=fair B=poor *OR* ρ ≥ 0**



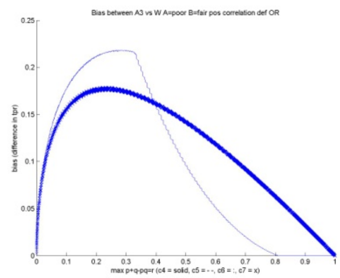

**A=fair B=poor *OR* unequal priors ρ ≥ 0**



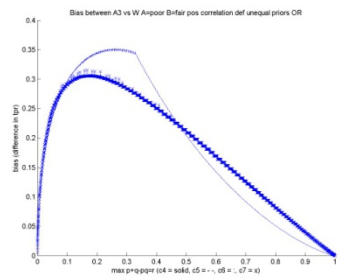

**A=poor B=good *OR* ρ ≥ 0**



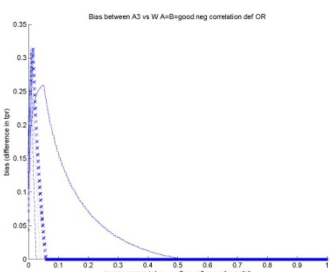

**A=poor B=good *OR* unequal priors ρ ≥ 0**



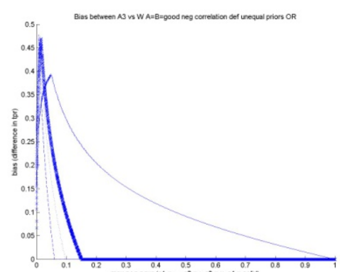

**A=poor B=fair *OR* ρ ≥ 0**



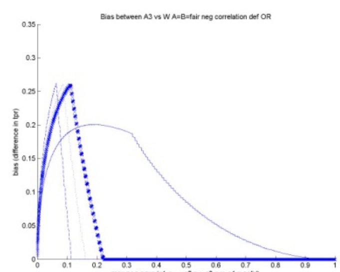

**A=poor B=fair *OR* unequal priors ρ ≥ 0**



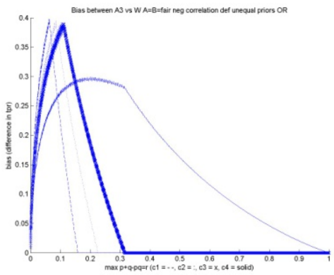

**A=B=good *OR* ρ ≤ 0**



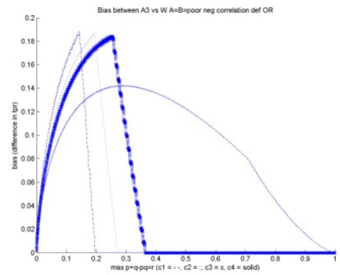

**A=B=good *OR* unequal priors ρ ≤ 0**



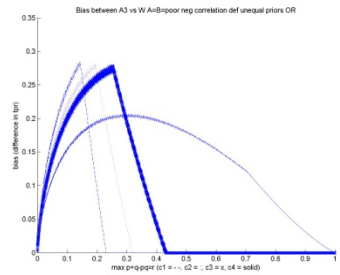

**A=B=fair *OR* ρ ≤ 0**



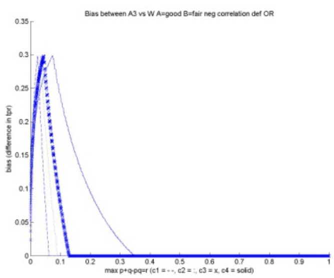

**A=B=fair *OR* unequal priors ρ ≤ 0**



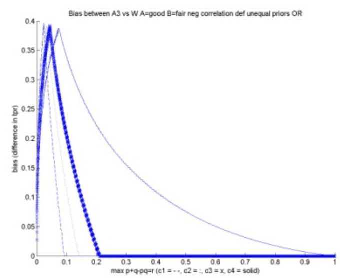

**A=B=poor *OR* ρ ≤ 0**



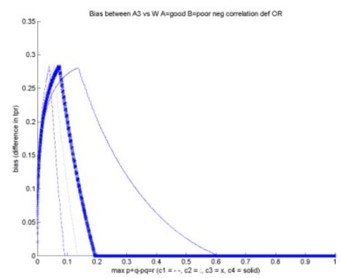

**A=B=poor unequal priors *OR* ρ ≤ 0**



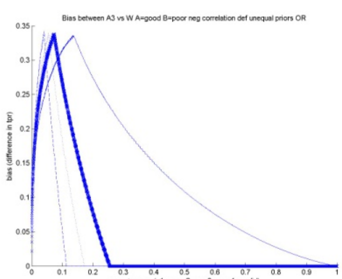

**A=good B=fair *OR* ρ ≤ 0**



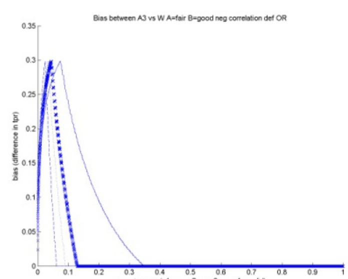

**A=good B=fair *OR* unequal priors ρ ≤ 0**



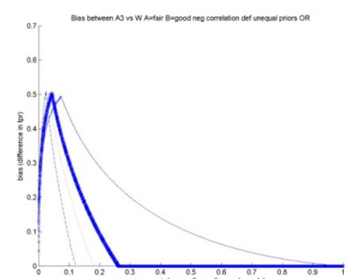

**A=good B=poor *OR* ρ ≤ 0**



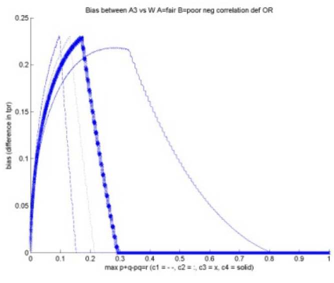

**A=good B=poor *OR* unequal priors ρ ≤ 0**



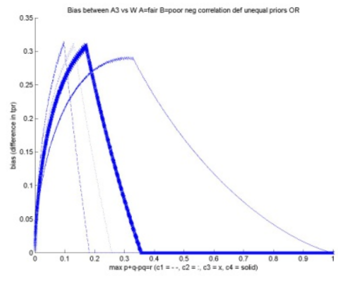

**A=fair B=good *OR* ρ ≤ 0**



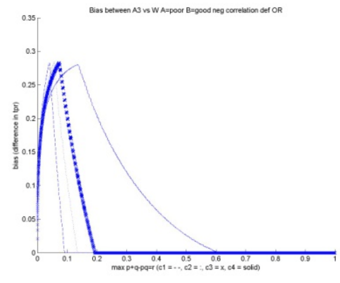

**A=fair B=good *OR* unequal priors ρ ≤ 0**



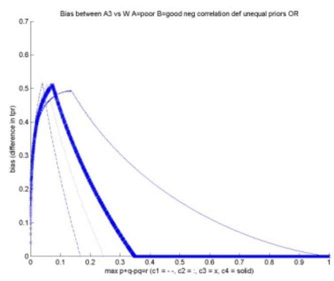

A=fair B=poor *OR* ρ ≤ 0



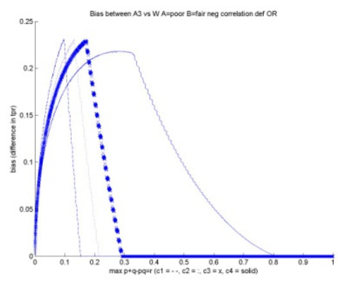

A=fair B=poor *OR* unequal priors ρ ≤ 0



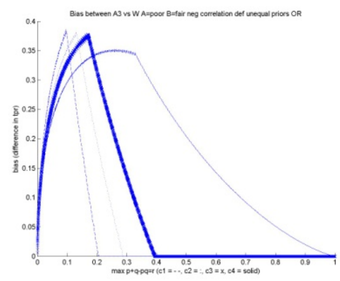

A=poor B=good *OR* ρ ≤ 0




A=poor B=good *OR* unequal priors ρ ≤ 0




A=poor B=fair *OR* ρ ≤ 0




A=poor B=fair *OR* unequal priors ρ ≤ 0

## *Appendix C Code*

Included in this appendix is all of the code used to run the simulation. First, a brief discussion of the algorithms used to create the simulation data is given. Following this section is the actual source code itself.

### *C.1 Outline of Program*

#### *C.1.1 Across I simulation*

The following contains a brief outline of the algorithm that has been developed to generate the ROC curves of interest for *across I* combined classification systems. In the text below, those items written in `courier new` refer to either variables or functions within the code itself.

1) Create the parameters for systems $A_\theta$ and $B_{\gamma,\varepsilon}$. Using *Matlab*, theta, gamma, and epsilon are created as linearly spaced row vectors. Each entry in the vector corresponds to a different threshold value for the parameter. At the max threshold, the accumulated probability is approximately one and for the minimum threshold, the accumulated probability is approximately zero.
2) Using the cumulative density function (CDF) for the normal distribution, the true positive and false positive rates are calculated for individual systems $A_\theta$ and $B_{\gamma,\varepsilon}$. For each threshold, a different probability measurement is assigned. System $B_{\gamma,\varepsilon}$ is created in such a way that the convex hull of the ROC curve must be found whereas classifier $A_\theta$ naturally produces a proper ROC curve. Refer to the function `rocB` for an explanation as to how the frontier of the ROC curve for system $B_{\gamma,\varepsilon}$ is created.

3) Feeding these values into the function `roc` computes and returns the combined Boolean *AND* ROC curve for systems $A_\theta$ and $B_{\gamma,\varepsilon}$.

4) Using the same information as generated in steps 1) and 2), inputting this data into the function `rocor` returns the combined Boolean *OR* ROC curve.

*C.1.2 Across II Simulation*

One interesting thing to note at this time is the following: regardless of the constitution of the targets that systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are designed to classify, the performance of the system is only based on two variables: target/non-target mean and target/non-target prevalence. Hence, the function for the true positive rate for system $A_\theta$ under *across I* label fusion is identical to the true positive rate for system $A_\theta$ under *across II* label fusion assuming the target prevalence and target means are the same. This same arguments holds for classifier system $B_{\gamma,\varepsilon}$ as well as *across III* and *within* label fusion rules. A brief outline of the *across II* simulation is now provided.

1) Load the same parameters (theta, gamma, epsilon) and ROC curves for classification systems $A_\theta$ and $B_{\gamma,\varepsilon}$ that were used for the combined *across I* simulation. The same data must be used in order to compare the performance of systems under different label-fusion assumptions.

2) Use the true positive and false positive rates from the ROC curves of classifier systems $A_\theta$ and $B_{\gamma,\varepsilon}$ as input for the functions `a2roc/a2rocor` that returns the combined *across II AND* and *across II OR* ROC curves for specified levels of performance for systems $A_\theta$ and $B_{\gamma,\varepsilon}$.

The algorithm remains approximately the same, but the weighting on the cost functions for classifier $A_\theta$ and $B_{\gamma,\varepsilon}$ are different as is outlined in section III.

*C.1.3 – Across III Simulation*

As discussed in section 3, in an *across III* label fusion environment, classifier system $A_\theta$ is tuned to classify the set union of $t_1$ and $t_2$ and classifier system $B_{\gamma,\varepsilon}$ is tuned to classify the set union of $t_2$ and $t_3$. As the targets for systems $A_\theta$ and $B_{\gamma,\varepsilon}$ are set unions, the calculation of true positive and false positive rates remain the same as the function depends only on parameter setting and target population mean. Recalling the cost functions outlined in section 3, the cost function for classifier $A_\theta$ is weighted by the prior probability of observing an inclusive $t_1$ or $t_2$ event and the prior probability of observing a non-target or $t_3$ event. The cost function for classifier $B_{\gamma,\varepsilon}$ is weighted by the prior probability of observing either a $t_2$ or $t_3$ event and the prior probability of observing a non-target or $t_1$ event.

The following encompasses a brief outline of the *across III* algorithm.

1) Load the same parameters (theta, gamma, epsilon) and ROC curves for classifiers $A_\theta$ and $B_{\gamma,\varepsilon}$ that were used for the combined *across I* simulation. The same data must be used in order to compare the performance of systems under different assumptions.
2) Load this information into the `roc` function to produce the ROC curves for the combined *AND* system and the `rocor` function for the combined *OR* ROC curve. The same function is used for *across III* as the input structure is identical to that of *across I* except that the prior probability weightings on the true positive and false positive rates for classifiers $A_\theta$ and $B_{\gamma,\varepsilon}$ is different.

The only difference in algorithm between *across I* and *across III* lies in the weighting of target/non-targets events as determined by the cost functions.

*C.1.4 Within Simulation*

Recall from section 3 that under the *within* label fusion environment, systems $A_\theta$ and $B_\Pi$ are tuned to the same target features. In this case, there is no distinction between target types. Interestingly, the calculations for the true positive and false positive rates of the individual systems is still the same due to the function relying only on parameter setting and target/non-target mean. The cost functions for the *within AND/OR* ROC curves is different from any of the *across* cost functions in that the cost function for each classifier is weighted only by total prior target prevalence and total prior non-target prevalence. Because of this, the performance of the combined *within* combined classification system is considerably better than the performance of any combined *across* classification system. Most, if not all, classification schemes popular in literature assume that all targets and non-targets share the same features and can be grouped into the *within* hierarchy of classifiers (*N.J. Leap et al: 2008).*

Below is a short description of the *within* algorithm.

1) Load the same parameters (theta, gamma, epsilon) and ROC curves for classification systems $A_\theta$ and $B_\Pi$ that were used in previous simulations. The same data must be used in order to compare the performance of systems under different assumptions.
2) Load this information into the `wroc/wrocor` functions to produce the ROC curves for the combined *AND/OR* systems.

*C.2 Source Code*

The code is presented here and any functions created by the author are explained following the main body of the code. Those items written in green are comments issued by the author that will assist in understanding the purpose of the code.

```
%% Classification systems Aθ and Bγ,ε are created here. Note that the
round function is called to halt the level of computational precision.

theta=linspace(-4,6,301);
gamma=theta;
epsilon=linspace(0,10,301);
n=length(theta);
save('parameters','theta','gamma','epsilon','n');

for i=1:n
    tpag(i)=round(probA(theta(i),0.95)*10000)/10000;
    tpaf(i)=round(probA(theta(i),0.85)*10000)/10000;
    tpap(i)=round(probA(theta(i),0.75)*10000)/10000;
    fpa(i)=round(1/2*erfc(theta(i)/sqrt(2))*10000)/10000;
end
save('Afront','tpag','tpaf','tpap','fpA');

for i=1:n
    for j=1:n
        tpBg(i,j)=round(probB(gamma(i),epsilon(j),0.95)*10000)/10000;
        tpBf(i,j)=round(probB(gamma(i),epsilon(j),0.85)*10000)/10000;
        tpBp(i,j)=round(probB(gamma(i),epsilon(j),0.75)*10000)/10000;
        fpB(i,j)=round(1/2*(erf((gamma(i)+epsilon(j))/sqrt(2))-
erf(gamma(i)/sqrt(2)))*10000)/10000;
    end
end

% B frontier
ubg=rocB(tpBg,fpB);
ubf=rocB(tpBf,fpB);
ubp=rocB(tpBp,fpB);
save('Bfront','ubg','ubf','ubp')

%% Within AND/OR all prevalence
%% A few things to note. First, you may read the variable “wgc1” as
“Within combined AND ROC curve A=good performance, B=good performance,
correlation P1 = -0.8 (as outlined in section V). Likewise, “wfgc1” is
read “Within combined AND ROC curve A=fair performance, B=poor
performance, and correlation P1 = -0.8.  If a c is not specified with a
variable, this implies that P = 0 (such as in the case where i>3 &
i<=4).

tic % begins computation timer.
load('Afront')
```

```
load('Bfront')

fpbg=ubg(:,1);
tpbg=ubg(:,2);
fpbf=ubf(:,1);
tpbf=ubf(:,2);
fpbp=ubp(:,1);
tpbp=ubp(:,2);

i=1;
while i<=7
if i<=1
load('tenhancedeq') % Loads the prior probabilities for the target
enhanced environment
wgc1=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfc1=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpc1=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfc1=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpc1=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgc1=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpc1=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgc1=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfc1=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wenc1','wgc1','wfc1','wpc1','wgfc1','wgpc1','wfgc1','wfpc1'
,'wpgc1','wpfc1') % The save destination is arbitrary.

wgorc1=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wforc1=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wporc1=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgforc1=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgporc1=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgorc1=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfporc1=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgorc1=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpforc1=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wenc1or','wgorc1','wforc1','wporc1','wgforc1','wgporc1','wf
gorc1','wfporc1','wpgorc1','wpforc1')

% Target rich environment
load('tricheq') % Loads the prior probabilities for the target rich
environment
wgrc1=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfrc1=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wprc1=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfrc1=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgprc1=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgrc1=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfprc1=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgrc1=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfrc1=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
```

```
save('C:\users\owner\desktop\within
files\wdata\wrichc1','wgrc1','wfrc1','wprc1','wgfrc1','wgprc1','wfgrc1'
,'wfprc1','wpgrc1','wpfrc1')

wgorrc1=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wforrc1=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wporrc1=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgforrc1=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgporrc1=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgorrc1=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfporrc1=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgorrc1=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpforrc1=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wrichc1or','wgorrc1','wforrc1','wporrc1','wgforrc1','wgporr
c1','wfgorrc1','wfporrc1','wpgorrc1','wpforrc1')

% Target deficient environment
load('tdefeq') %Loads the prior probabilities for the target deficient
environment
wgdc1=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfdc1=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpdc1=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfdc1=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpdc1=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgdc1=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpdc1=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgdc1=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfdc1=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wdefc1','wgdc1','wfdc1','wpdc1','wgfdc1','wgpdc1','wfgdc1',
'wfpdc1','wpgdc1','wpfdc1')

wgordc1=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfordc1=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpordc1=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfordc1=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpordc1=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgordc1=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpordc1=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgordc1=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfordc1=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wdefc1or','wgordc1','wfordc1','wpordc1','wgfordc1','wgpordc
1','wfgordc1','wfpordc1','wpgordc1','wpfordc1')
end

if i>1 && i<=2
load('tenhancedeq')
wgc2=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfc2=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpc2=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfc2=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpc2=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
```

```
wfgc2=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpc2=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgc2=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfc2=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wenc2','wgc2','wfc2','wpc2','wgfc2','wgpc2','wfgc2','wfpc2'
,'wpgc2','wpfc2')

wgorc2=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wforc2=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wporc2=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgforc2=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgporc2=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgorc2=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfporc2=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgorc2=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpforc2=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wenc2or','wgorc2','wforc2','wporc2','wgforc2','wgporc2','wf
gorc2','wfporc2','wpgorc2','wpforc2')

% Target rich environment
load('tricheq')
wgrc2=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfrc2=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wprc2=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfrc2=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgprc2=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgrc2=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfprc2=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgrc2=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfrc2=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wrichc2','wgrc2','wfrc2','wprc2','wgfrc2','wgprc2','wfgrc2'
,'wfprc2','wpgrc2','wpfrc2')

wgorrc2=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wforrc2=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wporrc2=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgforrc2=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgporrc2=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgorrc2=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfporrc2=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgorrc2=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpforrc2=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wrichc2or','wgorrc2','wforrc2','wporrc2','wgforrc2','wgporr
c2','wfgorrc2','wfporrc2','wpgorrc2','wpforrc2')

% Target deficient environment
load('tdefeq')
wgdc2=wroc(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfdc2=wroc(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpdc2=wroc(pet,pen,i,fpa,tpap,fpbp,tpbp);
```

```
wgfdc2=wroc(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpdc2=wroc(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgdc2=wroc(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpdc2=wroc(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgdc2=wroc(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfdc2=wroc(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wdefc2','wgdc2','wfdc2','wpdc2','wgfdc2','wgpdc2','wfgdc2',
'wfpdc2','wpgdc2','wpfdc2')

wgordc2=wrocor(pet,pen,i,fpa,tpag,fpbg,tpbg);
wfordc2=wrocor(pet,pen,i,fpa,tpaf,fpbf,tpbf);
wpordc2=wrocor(pet,pen,i,fpa,tpap,fpbp,tpbp);
wgfordc2=wrocor(pet,pen,i,fpa,tpag,fpbf,tpbf);
wgpordc2=wrocor(pet,pen,i,fpa,tpag,fpbp,tpbp);
wfgordc2=wrocor(pet,pen,i,fpa,tpaf,fpbg,tpbg);
wfpordc2=wrocor(pet,pen,i,fpa,tpaf,fpbp,tpbp);
wpgordc2=wrocor(pet,pen,i,fpa,tpap,fpbg,tpbg);
wpfordc2=wrocor(pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\within
files\wdata\wdefc2or','wgordc2','wfordc2','wpordc2','wgfordc2','wgpordc
2','wfgordc2','wfpordc2','wpgordc2','wpfordc2')
end
%% And so on and so forth up through i = 7.

function [Asys] = probA(x,y)
%Probability for classifier A
%Ni calculates the target/non-target mean of the system by the taking
the norm inverse of the AUC = y and multiplying the by the square root
of 2.
%Asys returns the probability of a true positive at a given x = theta
%parameter minus the mean.
%"erfc" stands for "error function complement"
Ni=norminv(y)*sqrt(2);
Asys=1/2*erfc((x-Ni)/sqrt(2));
end

function [Bsys] = probB(x,y,z)
%Finding TP/FP value for classifier B
%Ni is short for norm inverse of a given AUC value
%Bsys approximates the solution to the Gaussian integral for bounds
gamma
%to gamma plus epsilon
%"erf" stands for "error function"
Ni=norminv(z)*sqrt(2);
Bsys=1/2*(erf(((x+y)-Ni)/sqrt(2))-erf((x-Ni)/sqrt(2)));
end

function [rocB] = rocB(fp,tp)
% This function is used to find the ROC curve for classification system
%B.First, input variable "fp" = false positive rates and input variable
%"tp" = true positive rates. Take these NxN matrices and reshape them
%into (N^2)x1 column vectors and then sort by the false positive
%values. Take the difference of each (i,j) - ((i-1),(j-1)) entry in the
```

```
%true postive col Instead of looping over all values, this can be
%represented as a matrix operation by setting the first entry of the
%"difference matrix" to zero (therefore keeping the (1,1) entry) and
%shifting all of the remaining entries down by 1. Take the difference
%of the original TP col with this column. Now, find the indices where
%the difference (i,j) - ((i-1),(j-1)) < 0. These values are replaced by
%the preceding entry so that the true positive rate is always
%increasing. Take the unique rows. The function "rocheck" determines
%whether or not there are any remaining (i,j) - ((i-1),(j-1)) < 0, if
%so, run lines 17-33 again until rocheck returns the logical "1" =
%proper ROC curve. Starting at vfp, this part of function chooses the
%max tpr for each fpr (useful if length(rocB)is computationally
%unfeasible).

roc=sortrows(cat(2,fp(:),tp(:)),1);
c2=roc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=roc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
roc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

rocB=unique(roc,'rows');

i=1;
while i<=length(rocB);
if rocheck(rocB)~=1;
    rocB=cloop(rocB);
else if rocheck(rocB)==1;
        break
    end
end
i=i+1;
end

vfp=rocB(:,1);
vtp=rocB(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
rocB=unique(cat(2,fpmin',tpmax'),'rows');
end
```

```
function [wroc] = wroc(a,b,g,w,x,y,z)
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
                            rho=0.8;
                        end
                    end
                end
              end
          end
      end
end
n=length(w);
m=length(y);
% Cost functions for within along with correlation term.
for i=1:n
    for j=1:m
        fpand(i,j)=w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=a*z(j)+b*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpand(i,j)=(1/a)*(rho*corr(i,j)+costA(i)*costB(j)-
b*fpand(i,j));
    end
end
% Under certain correlations, the max(tpr) > 1, this set the upper
% bound at 1.
improc=sortrows(cat(2,fpand(:),tpand(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;
```

```
c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

wroc=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=length(wroc) %usually takes about 3000 iterations before
following step can assure a proper ROC curve for the size of this data
set.
if rocheck(wroc)~=1;
    wroc=cloop(wroc);
else if rocheck(wroc)==1;
        break
    end
end
i=i+1;
end

vfp=wroc(:,1);
vtp=wroc(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
wroc=unique(cat(2,fpmin',tpmax'),'rows');
end



function [wrocor] = wrocor(a,b,g,w,x,y,z)
% Almost exactly the same as wroc, but cost functions are different.
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
```

```
                            rho=0.8;
                        end
                    end
                end
              end
          end
    end
end
n=length(w);
m=length(y);
for i=1:n
    for j=1:m
        fpor(i,j)=w(i)+y(j)-w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=a*z(j)+b*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpor(i,j)=(1/a)*(costA(i)+costB(j)-rho*corr(i,j)-
(costA(i)*costB(j))-b*fpor(i,j));
    end
end

improc=sortrows(cat(2,fpor(:),tpor(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

wrocor=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=2000
```

```
if rocheck(wrocor)~=1;
    wrocor=cloop(wrocor);
else if rocheck(wrocor)==1;
        break
    end
end
i=i+1;
end

vfp=wrocor(:,1);
vtp=wrocor(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
wrocor=unique(cat(2,fpmin',tpmax'),'rows');
end

function [cloop] = cloop(x)
%cloop performs the same "difference" function as outlined in all roc
%functions
cloop=cat(2,x(:,1),x(:,2));
c1=cloop(:,1);
c2=cloop(:,2);
q=length(cloop);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
cloop(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end
cloop=unique(round(cloop*10000)/10000,'rows');
end

function [rocheck] = rocheck(x)
%Checks to make sure that all true positive rates are strictly
increasing.
%It returns a "1" if this condition is true and a "0" if this condition
is
%false.
q=length(x);
c2=x(:,2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

if diff>=0;
    rocheck=1;
```

```
else
    rocheck=0;
end
end

% Across II all prev
tic
load('Afront')
load('Bfront')
% Target enhanced environment
fpbg=ubg(:,1);
tpbg=ubg(:,2);
fpbf=ubf(:,1);
tpbf=ubf(:,2);
fpbp=ubp(:,1);
tpbp=ubp(:,2);

i=1;
while i<=7
if i<=1
load('tena2')
a2gc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2fc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2pc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gfc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gpc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
a2fgc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fpc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pfc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2enc1','a2gc1','a2fc1','a2pc1','a2gfc1','a2gpc1','a2fgc1'
,'a2fpc1','a2pgc1','a2pfc1')

a2gorc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2forc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2porc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gforc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gporc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
a2fgorc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fporc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgorc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pforc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2enc1or','a2gorc1','a2forc1','a2porc1','a2gforc1','a2gpor
c1','a2fgorc1','a2fporc1','a2pgorc1','a2pforc1')

% Target rich environment
load('tricha2')
a2grc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2frc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2prc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gfrc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gprc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
```

```
a2fgrc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fprc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgrc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pfrc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2richc1','a2grc1','a2frc1','a2prc1','a2gfrc1','a2gprc1','
a2fgrc1','a2fprc1','a2pgrc1','a2pfrc1')

a2gorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2forrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2porrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gforrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gporrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
a2fgorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fporrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pforrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2richc1or','a2gorrc1','a2forrc1','a2porrc1','a2gforrc1','
a2gporrc1','a2fgorrc1','a2fporrc1','a2pgorrc1','a2pforrc1')

% Target deficient environment
load('tdefa2')
a2gdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2fdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2pdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gfdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gpdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
a2fgdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fpdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pfdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2defc1','a2gdc1','a2fdc1','a2pdc1','a2gfdc1','a2gpdc1','a
2fgdc1','a2fpdc1','a2pgdc1','a2pfdc1')

a2gordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
a2fordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a2pordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
a2gfordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
a2gpordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
a2fgordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a2fpordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a2pgordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
a2pfordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\a2defc1or','a2gordc1','a2fordc1','a2pordc1','a2gfordc1','a
2gpordc1','a2fgordc1','a2fpordc1','a2pgordc1','a2pfordc1')
end
% We will abbreviate this here as it just continues on through the
remainder of the possible values of the correlation coefficient.

% Across II all prev unequal target priors. In much the same spirit as
above, this is an example of the code where we are solving for the ROC
```

```
curves after altering the prevalence of targets in the Across II
environment.

tic
load('Afront')
load('Bfront')
% Target enhanced environment
fpbg=ubg(:,1);
tpbg=ubg(:,2);
fpbf=ubf(:,1);
tpbf=ubf(:,2);
fpbp=ubp(:,1);
tpbp=ubp(:,2);

i=1;
while i<=7
if i<=1
load('altpreva2')
aa2gc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2fc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2pc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gfc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gpc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fpc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa2pfc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2enc1','aa2gc1','aa2fc1','aa2pc1','aa2gfc1','aa2gpc1','a
a2fgc1','aa2fpc1','aa2pgc1','aa2pfc1')

aa2gorc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2forc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2porc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gforc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gporc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgorc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fporc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgorc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa2pforc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2enc1or','aa2gorc1','aa2forc1','aa2porc1','aa2gforc1','a
a2gporc1','aa2fgorc1','aa2fporc1','aa2pgorc1','aa2pforc1')

% Target rich environment
load('apricha2')
aa2grc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2frc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2prc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gfrc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gprc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgrc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fprc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgrc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
```

```
aa2pfrc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2richc1','aa2grc1','aa2frc1','aa2prc1','aa2gfrc1','aa2gp
rc1','aa2fgrc1','aa2fprc1','aa2pgrc1','aa2pfrc1')

aa2gorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2forrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2porrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gforrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gporrc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fporrc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgorrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa2pforrc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2richc1or','aa2gorrc1','aa2forrc1','aa2porrc1','aa2gforr
c1','aa2gporrc1','aa2fgorrc1','aa2fporrc1','aa2pgorrc1','aa2pforrc1')

% Target deficient environment
load('apdefa2')
aa2gdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2fdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2pdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gfdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gpdc1=roc2(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fpdc1=roc2(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa2pfdc1=roc2(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2defc1','aa2gdc1','aa2fdc1','aa2pdc1','aa2gfdc1','aa2gpd
c1','aa2fgdc1','aa2fpdc1','aa2pgdc1','aa2pfdc1')

aa2gordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa2fordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa2pordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa2gfordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa2gpordc1=roc2or(petA,penA,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa2fgordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa2fpordc1=roc2or(petA,penA,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa2pgordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa2pfordc1=roc2or(petA,penA,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('c:\users\owner\desktop\a2
files\a2data\aa2defc1or','aa2gordc1','aa2fordc1','aa2pordc1','aa2gfordc
1','aa2gpordc1','aa2fgordc1','aa2fpordc1','aa2pgordc1','aa2pfordc1')
end
% And so on…

function [roc2] = roc2(a,b,c,d,g,w,x,y,z)
% roc function for across II AND
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
```

```
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
                            rho=0.8;
                        end
                    end
                end
              end
          end
      end
end
n=length(w);
m=length(y);
for i=1:n
    for j=1:m
        fpand(i,j)=w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=c*z(j)+d*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpand(i,j)=(1/c)*(rho*corr(i,j)+costA(i)*costB(j)-
d*fpand(i,j));
    end
end

improc=sortrows(cat(2,fpand(:),tpand(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
```

```
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

roc2=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=length(roc2)
if rocheck(roc2)~=1;
    roc2=cloop(roc2);
else if rocheck(roc2)==1;
        break
    end
end
i=i+1;
end

vfp=roc2(:,1);
vtp=roc2(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
roc2=unique(cat(2,fpmin',tpmax'),'rows');
end

function [roc2or] = roc2or(a,b,c,d,g,w,x,y,z)
% function for across II OR ROC
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
                            rho=0.8;
                        end
                    end
                end
              end
          end
    end
end
n=length(w);
m=length(y);
```

```
for i=1:n
    for j=1:m
        fpor(i,j)=w(i)+y(j)-w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=c*z(j)+d*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpor(i,j)=(1/c)*(costA(i)+costB(j)-rho*corr(i,j)-
(costA(i)*costB(j))-d*fpor(i,j));
    end
end

improc=sortrows(cat(2,fpor(:),tpor(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

roc2or=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=length(roc2or);
if rocheck(roc2or)~=1;
    roc2or=cloop(roc2or);
else if rocheck(roc2or)==1;
        break
    end
end
i=i+1;
end

vfp=roc2or(:,1);
```

```
vtp=roc2or(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
roc2or=unique(cat(2,fpmin',tpmax'),'rows');
end
```

Note: The following section of code works for both *across I* and *across III*, only the input is different. This general algorithm follows the same outline as before.

```
% Across III all prev
tic
load('Afront')
load('Bfront')

fpbg=ubg(:,1);
tpbg=ubg(:,2);
fpbf=ubf(:,1);
tpbf=ubf(:,2);
fpbp=ubp(:,1);
tpbp=ubp(:,2);

i=1;
while i<=7
if i<=1
load('tena3')
a3gc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3fc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3pc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gfc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gpc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fpc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a3pgc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pfc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\a3enc1','a3gc1','a3fc1','a3pc1','a3gfc1','a3gpc1','a3fgc1'
,'a3fpc1','a3pgc1','a3pfc1')

a3gorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3forc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3porc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gforc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gporc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fporc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a3pgorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pforc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
```

```
save('C:\users\owner\desktop\A3
files\a3data\a3enc1or','a3gorc1','a3forc1','a3porc1','a3gforc1','a3gpor
c1','a3fgorc1','a3fporc1','a3pgorc1','a3pforc1')

% Target rich environment
load('tricha3')
a3grc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3frc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3prc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gfrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gprc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fprc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a3pgrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pfrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\a3richc1','a3grc1','a3frc1','a3prc1','a3gfrc1','a3gprc1','
a3fgrc1','a3fprc1','a3pgrc1','a3pfrc1')

a3gorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3forrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3porrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gforrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gporrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fporrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a3pgorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pforrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\a3richc1or','a3gorrc1','a3forrc1','a3porrc1','a3gforrc1','
a3gporrc1','a3fgorrc1','a3fporrc1','a3pgorrc1','a3pforrc1')

% Target deficient environment
load('tdefa3')
a3gdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3fdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3pdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gfdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gpdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fpdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
a3pgdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pfdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\a3defc1','a3gdc1','a3fdc1','a3pdc1','a3gfdc1','a3gpdc1','a
3fgdc1','a3fpdc1','a3pgdc1','a3pfdc1')

a3gordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
a3fordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
a3pordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
a3gfordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
a3gpordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
a3fgordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
a3fpordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
```

```
a3pgordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
a3pfordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\a3defc1or','a3gordc1','a3fordc1','a3pordc1','a3gfordc1','a
3gpordc1','a3fgordc1','a3fpordc1','a3pgordc1','a3pfordc1')
end
% Up to i=7 and just for kicks we'll look at one iteration of the loop
that solves for those ROC curves where the prevalence of t12 and t23 are
altered under the across III label fusion rule.

% Across III all prev
tic
load('Afront')
load('Bfront')

fpbg=ubg(:,1);
tpbg=ubg(:,2);
fpbf=ubf(:,1);
tpbf=ubf(:,2);
fpbp=ubp(:,1);
tpbp=ubp(:,2);

i=1;
while i<=7
if i<=1
load('altpreva3')
aa3gc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3fc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3pc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gfc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gpc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fpc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pfc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3enc1','aa3gc1','aa3fc1','aa3pc1','aa3gfc1','aa3gpc1','a
a3fgc1','aa3fpc1','aa3pgc1','aa3pfc1')

aa3gorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3forc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3porc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gforc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gporc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fporc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgorc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pforc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3enc1or','aa3gorc1','aa3forc1','aa3porc1','aa3gforc1','a
a3gporc1','aa3fgorc1','aa3fporc1','aa3pgorc1','aa3pforc1')

% Target rich environment
load('apricha3')
```

```
aa3grc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3frc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3prc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gfrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gprc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fprc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pfrc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3richc1','aa3grc1','aa3frc1','aa3prc1','aa3gfrc1','aa3gp
rc1','aa3fgrc1','aa3fprc1','aa3pgrc1','aa3pfrc1')

aa3gorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3forrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3porrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gforrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gporrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fporrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgorrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pforrc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3richc1or','aa3gorrc1','aa3forrc1','aa3porrc1','aa3gforr
c1','aa3gporrc1','aa3fgorrc1','aa3fporrc1','aa3pgorrc1','aa3pforrc1')

% Target deficient environment
load('apdefa3')
aa3gdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3fdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3pdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gfdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gpdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fpdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pfdc1=roc(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3defc1','aa3gdc1','aa3fdc1','aa3pdc1','aa3gfdc1','aa3gpd
c1','aa3fgdc1','aa3fpdc1','aa3pgdc1','aa3pfdc1')

aa3gordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbg,tpbg);
aa3fordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbf,tpbf);
aa3pordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbp,tpbp);
aa3gfordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbf,tpbf);
aa3gpordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpag,fpbp,tpbp);
aa3fgordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbg,tpbg);
aa3fpordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpaf,fpbp,tpbp);
aa3pgordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbg,tpbg);
aa3pfordc1=rocor(pet12,pen12,pet23,pen23,pet,pen,i,fpa,tpap,fpbf,tpbf);
save('C:\users\owner\desktop\A3
files\a3data\aa3defc1or','aa3gordc1','aa3fordc1','aa3pordc1','aa3gfordc
1','aa3gpordc1','aa3fgordc1','aa3fpordc1','aa3pgordc1','aa3pfordc1')
end
```

```
function [roc] = roc(a,b,c,d,e,f,g,w,x,y,z)
% roc function for across I/ across III AND. Note the only difference
is
% the prior target weightings.
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
                            rho=0.8;
                        end
                    end
                end
              end
          end
      end
end
n=length(w);
m=length(y);
for i=1:n
    for j=1:m
        fpand(i,j)=w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=c*z(j)+d*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpand(i,j)=(1/e)*(rho*corr(i,j)+costA(i)*costB(j)-
f*fpand(i,j));
    end
end

improc=sortrows(cat(2,fpand(:),tpand(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
```

```
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

roc=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=length(roc)
if rocheck(roc)~=1;
    roc=cloop(roc);
else if rocheck(roc)==1;
        break
    end
end
i=i+1;
end

vfp=roc(:,1);
vtp=roc(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
roc=unique(cat(2,fpmin',tpmax'),'rows');
end

function [rocor] = rocor(a,b,c,d,e,f,g,w,x,y,z)
% function that produces across I/ across III OR ROC curves
if g==1
    rho=-0.8;
else if g==2
        rho=-0.5;
    else if g==3
            rho=-0.3;
        else if g==4
                rho=0;
            else if g==5
                    rho=0.3;
                else if g==6
                        rho=0.5;
                    else if g==7
                            rho=0.8;
```

```
                        end
                    end
                end
              end
          end
    end
end
n=length(w);
m=length(y);
for i=1:n
    for j=1:m
        fpor(i,j)=w(i)+y(j)-w(i)*y(j);
        costA(i)=a*x(i)+b*w(i);
        costB(j)=c*z(j)+d*y(j);
        corr(i,j)=sqrt(costA(i)*(1-costA(i)))*sqrt(costB(j)*(1-
costB(j)));
        tpor(i,j)=(1/e)*(costA(i)+costB(j)-rho*corr(i,j)-
(costA(i)*costB(j))-f*fpor(i,j));
    end
end

improc=sortrows(cat(2,fpor(:),tpor(:)),1);
c2=improc(:,2);
btp=find(c2<0);
if isempty(btp)~=1;
    c2(btp)=0;
end

ind=find(c2>1);
if isempty(ind)~=1;
c2(ind)=1;
end
improc=cat(2,improc(:,1),c2);

c2=improc(:,2);
q=length(c2);
dmat=[0;c2(1:(q-1))];
diff=c2-dmat;

c1=improc(:,1);
nf=find(diff<0);
n=length(nf);

i=1;
while i<=n
improc(nf(i),:)=cat(2,c1(nf(i)),c2(nf(i)-1));
    i=i+1;
end

rocor=unique(round(improc*10000)/10000,'rows');

i=1;
while i<=length(rocor)
if rocheck(rocor)~=1;
```

```
    rocor=cloop(rocor);
else if rocheck(rocor)==1;
        break
    end
end
i=i+1;
end

vfp=rocor(:,1);
vtp=rocor(:,2);

for i=1:length(vfp)
    ind=find(vfp==vfp(i));
    tpmax(i)=max(vtp(ind));
    fpmin(i)=min(vfp(ind));
end
rocor=unique(cat(2,fpmin',tpmax'),'rows');
end

function [bias] = bias(x,y)
%Bias between within and across classifiers.
%c1 and c2 are the false positive and true positive rates for the ROC
curve
%of the across classifier
%c3 and c4 are the fpr and tpr for the ROC curve of the within
classifier
c1=x(:,1);
c2=x(:,2);
c3=y(:,1);
c4=y(:,2);
% Returns the indices of those fpr values that lie in the intersection
of
% the fpr for the across classifier and the fpr for the within
classifier
[r,a,w]=intersect(c1,c3);
adata=cat(2,r,c2(a)); % fpr and tpr for across classifier
wdata=cat(2,r,c4(w)); % fpr and tpr for within classifier
bias=cat(2,r,wdata(:,2)-adata(:,2)); % bias = difference in tpr at
given fpr
end

function [negbias] = negbias(a,b,c,d,e,f,s,h,q,r)
% Returns graphs for non-correlated and all negitively correlated ROC
% curves.

g=bias(a,b); %non-correlated
g2=bias(c,d); %c1
g3=bias(e,f); %c2
g4=bias(s,h); %c3

negbias=figure('visible','off');
hold on
plot(g(:,1),g(:,2))
plot(g2(:,1),g2(:,2),'--')
```

```
plot(g3(:,1),g3(:,2),':')
plot(g4(:,1),g4(:,2),'x')
xlabel('max pq=r')
ylabel('bias (difference in tpr)')
title(q)
saveas(negbias,r)
end

function [negbiasor] = negbiasor(a,b,c,d,e,f,s,h,q,r)
% Returns graphs for non-correlated and all negitively correlated ROC
% curves.

g=bias(a,b); %non-correlated
g2=bias(c,d); %c1
g3=bias(e,f); %c2
g4=bias(s,h); %c3

negbiasor=figure('visible','off');
hold on
plot(g(:,1),g(:,2))
plot(g2(:,1),g2(:,2),'--')
plot(g3(:,1),g3(:,2),':')
plot(g4(:,1),g4(:,2),'x')
xlabel('max p+q-pq=r')
ylabel('bias (difference in tpr)')
title(q)
saveas(negbiasor,r)
end

function [posbias] = posbias(a,b,c,d,e,f,s,h,q,r)
% Returns graphs for non-correlated and all positively correlated ROC
% curves.

g=bias(a,b); %non-correlated
g2=bias(c,d); %c1
g3=bias(e,f); %c2
g4=bias(s,h); %c3
posbias=figure('visible','off');
hold on
plot(g(:,1),g(:,2))
plot(g2(:,1),g2(:,2),'--')
plot(g3(:,1),g3(:,2),':')
plot(g4(:,1),g4(:,2),'x')
xlabel('max pq=r')
ylabel('bias (difference in tpr)')
title(q)
saveas(posbias,r)
end

function [posbiasor] = posbiasor(a,b,c,d,e,f,s,h,q,r)
% Returns graphs for non-correlated and all positively correlated ROC
% curves.

g=bias(a,b); %non-correlated
```

```
g2=bias(c,d); %c1
g3=bias(e,f); %c2
g4=bias(s,h); %c3
posbiasor=figure('visible','off');
hold on
plot(g(:,1),g(:,2))
plot(g2(:,1),g2(:,2),'--')
plot(g3(:,1),g3(:,2),':')
plot(g4(:,1),g4(:,2),'x')
xlabel('max p+q-pq=r')
ylabel('bias (difference in tpr)')
title(q)
saveas(posbiasor,r)
end
```

For brevity, only one section of the figures code is included here. All other codes are approximately the same except for changes in variables and strings.

```
clear;clc
%% Enhanced figures alt prev
load('a1en');
load('Bfront');
load('Afront');
load('a1enor');
load('wen');
load('wenor');
load('a2en');
load('a2enor');
load('a3en');
load('a3enor');
%% A vs B I & III
AvBg=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
xlabel('Classifier A Good performance = line')
ylabel('Classifier B Good performance = dash')
title('A vs B  "good"')
saveas(AvBg,'c:\users\owner\desktop\graphs\ETP\A vs B G.jpg')

AvBf=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
xlabel('Classifier A Good performance = line')
ylabel('Classifier B Good performance = dash')
title('A vs B  "fair"')
saveas(AvBf,'c:\users\owner\desktop\graphs\ETP\A vs B F.jpg')

AvBp=figure('visible','off');
hold on
```

```
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
xlabel('Classifier A Good performance = line')
ylabel('Classifier B Good performance = dash')
title('A vs B  "poor"')
saveas(AvBp,'c:\users\owner\desktop\graphs\ETP\A vs B P.jpg')

%% Across I vs within ROC
G1=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1g(:,1),a1g(:,2),':')
plot(wg(:,1),wg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "good" vs within "good"')
saveas(G1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I G vs
within G.jpg')

F1=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a1f(:,1),a1f(:,2),':')
plot(wf(:,1),wf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "fair" vs within "fair"')
saveas(F1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I F vs
within F.jpg')

P1=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1p(:,1),a1p(:,2),':')
plot(wp(:,1),wp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "poor" vs within "poor"')
saveas(P1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I P vs
within P.jpg')

gf1=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
plot(a1gf(:,1),a1gf(:,2),':')
plot(wgf(:,1),wgf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=G B=F vs within A=G B=F')
```

```
saveas(gf1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=G
B=F vs within A=G B=F.jpg')

gp1=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1gp(:,1),a1gp(:,2),':')
plot(wgp(:,1),wgp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=G B=P vs within A=G B=P')
saveas(gp1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=G
B=P vs within A=G B=P.jpg')

fg1=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1fg(:,1),a1fg(:,2),':')
plot(wfg(:,1),wfg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=F B=G vs within A=F B=G')
saveas(fg1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=F
B=G vs within A=F B=G.jpg')

fp1=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1fp(:,1),a1fp(:,2),':')
plot(wfp(:,1),wfp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=F B=P vs within A=F B=P')
saveas(fp1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=F
B=P vs within A=F B=P.jpg')

pg1=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1pg(:,1),a1pg(:,2),':')
plot(wpg(:,1),wpg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=P B=G vs within A=P B=G')
saveas(pg1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=P
B=G vs within A=P B=G.jpg')

pf1=figure('visible','off');
hold on
plot(fpa,ap)
```

```
plot(ubf(:,1),ubf(:,2),'--')
plot(a1pf(:,1),a1pf(:,2),':')
plot(wpf(:,1),wpf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=P B=F vs within A=P B=F')
saveas(pf1, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=P
B=F vs within A=P B=F.jpg')

%% ROC Across II vs within AND
G2=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2g(:,1),a2g(:,2),':')
plot(wg(:,1),wg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "good" vs within "good"')
saveas(G2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II G vs
within G.jpg')

F2=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2f(:,1),a2f(:,2),':')
plot(wf(:,1),wf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "fair" vs within "fair"')
saveas(F2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II F vs
within F.jpg')

P2=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2p(:,1),a2p(:,2),':')
plot(wp(:,1),wp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "poor" vs within "poor"')
saveas(P2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II P vs
within P.jpg')

GF2=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2gf(:,1),a2gf(:,2),':')
plot(wgf(:,1),wgf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
```

```
title('A vs B vs Across II A=G B=F vs within A=G B=F')
saveas(GF2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=G
B=F vs within A=G B=F.jpg')

GP2=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2gp(:,1),a2gp(:,2),':')
plot(wgp(:,1),wgp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=G B=P vs within A=G B=P')
saveas(GP2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=G
B=P vs within A=G B=P.jpg')

FG2=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2fg(:,1),a2fg(:,2),':')
plot(wfg(:,1),wfg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=F B=G vs within A=F B=G')
saveas(FG2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=F
B=G vs within A=F B=G.jpg')

FP2=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2fp(:,1),a2fp(:,2),':')
plot(wfp(:,1),wfp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=F B=P vs within A=F B=P')
saveas(FP2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=F
B=P vs within A=F B=P.jpg')

PG2=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2pg(:,1),a2pg(:,2),':')
plot(wpg(:,1),wpg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=P B=G vs within A=P B=G')
saveas(PG2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=P
B=G vs within A=P B=G.jpg')

PF2=figure('visible','off');
hold on
```

```
plot(fpa,ap)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2pf(:,1),a2pf(:,2),':')
plot(wpf(:,1),wpf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=P B=F vs within A=P B=F')
saveas(PF2, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II A=P
B=F vs within A=P B=F.jpg')

%% Across 3 vs within ROC
G3=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3g(:,1),a3g(:,2),':')
plot(wg(:,1),wg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "good" vs within "good"')
saveas(G3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III G vs
within G.jpg')

F3=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a3f(:,1),a3f(:,2),':')
plot(wf(:,1),wf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "fair" vs within "fair"')
saveas(F3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III F vs
within F.jpg')

P3=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3p(:,1),a3p(:,2),':')
plot(wp(:,1),wp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "poor" vs within "poor"')
saveas(P3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III P vs
within P.jpg')

gf3=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
plot(a3gf(:,1),a3gf(:,2),':')
plot(wgf(:,1),wgf(:,2),'*')
xlabel('A = solid, B = dash')
```

```
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=G B=F vs within A=G B=F')
saveas(gf3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=G
B=F vs within A=G B=F.jpg')

gp3=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3gp(:,1),a3gp(:,2),':')
plot(wgp(:,1),wgp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=G B=P vs within A=G B=P')
saveas(gp3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=G
B=P vs within A=G B=P.jpg')

fg3=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3fg(:,1),a3fg(:,2),':')
plot(wfg(:,1),wfg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=F B=G vs within A=F B=G')
saveas(fg3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=F
B=G vs within A=F B=G.jpg')

fp3=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3fp(:,1),a3fp(:,2),':')
plot(wfp(:,1),wfp(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=F B=P vs within A=F B=P')
saveas(fp3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=F
B=P vs within A=F B=P.jpg')

pg3=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3pg(:,1),a3pg(:,2),':')
plot(wpg(:,1),wpg(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=P B=G vs within A=P B=G')
saveas(pg3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=P
B=G vs within A=P B=G.jpg')

pf3=figure('visible','off');
```

```
hold on
plot(fpa,ap)
plot(ubf(:,1),ubf(:,2),'--')
plot(a3pf(:,1),a3pf(:,2),':')
plot(wpf(:,1),wpf(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=P B=F vs within A=P B=F')
saveas(pf3, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III A=P
B=F vs within A=P B=F.jpg')

%% ROC Across I vs within OR
G1or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1gor(:,1),a1gor(:,2),':')
plot(wgor(:,1),wgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "good" vs within "good" OR')
saveas(G1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I G vs
within G OR.jpg')

F1or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a1for(:,1),a1for(:,2),':')
plot(wfor(:,1),wfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "fair" vs within "fair" OR')
saveas(F1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I F vs
within F OR.jpg')

P1or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1por(:,1),a1por(:,2),':')
plot(wpor(:,1),wpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I "poor" vs within "poor" OR')
saveas(P1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I P vs
within P OR.jpg')

gf1or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
plot(a1gfor(:,1),a1gfor(:,2),':')
plot(wgfor(:,1),wgfor(:,2),'*')
```

```
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=G B=F vs within A=G B=F OR')
saveas(gf1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=G
B=F vs within A=G B=F OR.jpg')

gp1or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1gpor(:,1),a1gpor(:,2),':')
plot(wgpor(:,1),wgpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=G B=P vs within A=G B=P OR')
saveas(gp1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=G
B=P vs within A=G B=P OR.jpg')

fg1or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1fgor(:,1),a1fgor(:,2),':')
plot(wfgor(:,1),wfgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=F B=G vs within A=F B=G OR')
saveas(fg1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=F
B=G vs within A=F B=G OR.jpg')

fp1or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a1fpor(:,1),a1fpor(:,2),':')
plot(wfpor(:,1),wfpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=F B=P vs within A=F B=P OR')
saveas(fp1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=F
B=P vs within A=F B=P OR.jpg')

pg1or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a1pgor(:,1),a1pgor(:,2),':')
plot(wpgor(:,1),wpgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=P B=G vs within A=P B=G OR')
saveas(pg1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=P
B=G vs within A=P B=G OR.jpg')
```

```
pf1or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubf(:,1),ubf(:,2),'--')
plot(a1pfor(:,1),a1pfor(:,2),':')
plot(wpfor(:,1),wpfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A1 =  double dot, w = star')
title('A vs B vs Across I A=P B=F vs within A=P B=F OR')
saveas(pf1or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across I A=P
B=F vs within A=P B=F OR.jpg')

%% ROC II OR
G2or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2gor(:,1),a2gor(:,2),':')
plot(wgor(:,1),wgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "good" vs within "good" OR')
saveas(G2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II G
vs within G OR.jpg')

F2or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2for(:,1),a2for(:,2),':')
plot(wfor(:,1),wfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "fair" vs within "fair" OR')
saveas(F2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II F
vs within F OR.jpg')

P2or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2por(:,1),a2por(:,2),':')
plot(wpor(:,1),wpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II "poor" vs within "poor" OR')
saveas(P2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II P
vs within P OR.jpg')

gf2or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2gfor(:,1),a2gfor(:,2),':')
```

```
plot(wgfor(:,1),wgfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=G B=F vs within A=G B=F OR')
saveas(gf2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=G B=F vs within A=G B=F OR.jpg')

gp2or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2gpor(:,1),a2gpor(:,2),':')
plot(wgpor(:,1),wgpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=G B=P vs within A=G B=P OR')
saveas(gp2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=G B=P vs within A=G B=P OR.jpg')

fg2or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2fgor(:,1),a2fgor(:,2),':')
plot(wfgor(:,1),wfgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=F B=G vs within A=F B=G OR')
saveas(fg2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=F B=G vs within A=F B=G OR.jpg')

fp2or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a2fpor(:,1),a2fpor(:,2),':')
plot(wfpor(:,1),wfpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=F B=P vs within A=F B=P OR')
saveas(fp2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=F B=P vs within A=F B=P OR.jpg')

pg2or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a2pgor(:,1),a2pgor(:,2),':')
plot(wpgor(:,1),wpgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=P B=G vs within A=P B=G OR')
saveas(pg2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=P B=G vs within A=P B=G OR.jpg')
```

```
pf2or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubf(:,1),ubf(:,2),'--')
plot(a2pfor(:,1),a2pfor(:,2),':')
plot(wpfor(:,1),wpfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A2 = double dot, w = star')
title('A vs B vs Across II A=P B=F vs within A=P B=F OR')
saveas(pf2or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across II
A=P B=F vs within A=P B=F OR.jpg')

%% Roc Across III vs within OR
G3or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3gor(:,1),a3gor(:,2),':')
plot(wgor(:,1),wgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "good" vs within "good" OR')
saveas(G3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III G
vs within G OR.jpg')

F3or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubf(:,1),ubf(:,2),'--')
plot(a3for(:,1),a3for(:,2),':')
plot(wfor(:,1),wfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "fair" vs within "fair" OR')
saveas(F3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III F
vs within F OR.jpg')

P3or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3por(:,1),a3por(:,2),':')
plot(wpor(:,1),wpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III "poor" vs within "poor" OR')
saveas(P3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III P
vs within P OR.jpg')

gf3or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubf(:,1),ubf(:,2),'--')
```

```
plot(a3gfor(:,1),a3gfor(:,2),':')
plot(wgfor(:,1),wgfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=G B=F vs within A=G B=F OR')
saveas(gf3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=G B=F vs within A=G B=F OR.jpg')

gp3or=figure('visible','off');
hold on
plot(fpa,ag)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3gpor(:,1),a3gpor(:,2),':')
plot(wgpor(:,1),wgpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=G B=P vs within A=G B=P OR')
saveas(gp3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=G B=P vs within A=G B=P OR.jpg')

fg3or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3fgor(:,1),a3fgor(:,2),':')
plot(wfgor(:,1),wfgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=F B=G vs within A=F B=G OR')
saveas(fg3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=F B=G vs within A=F B=G OR.jpg')

fp3or=figure('visible','off');
hold on
plot(fpa,af)
plot(ubp(:,1),ubp(:,2),'--')
plot(a3fpor(:,1),a3fpor(:,2),':')
plot(wfpor(:,1),wfpor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=F B=P vs within A=F B=P OR')
saveas(fp3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=F B=P vs within A=F B=P OR.jpg')

pg3or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubg(:,1),ubg(:,2),'--')
plot(a3pgor(:,1),a3pgor(:,2),':')
plot(wpgor(:,1),wpgor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=P B=G vs within A=P B=G OR')
```

```
saveas(pg3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=P B=G vs within A=P B=G OR.jpg')

pf3or=figure('visible','off');
hold on
plot(fpa,ap)
plot(ubf(:,1),ubf(:,2),'--')
plot(a3pfor(:,1),a3pfor(:,2),':')
plot(wpfor(:,1),wpfor(:,2),'*')
xlabel('A = solid, B = dash')
ylabel('A3 =  double dot, w = star')
title('A vs B vs Across III A=P B=G vs within A=P B=G OR')
title('A vs B vs Across III A=P B=F vs within A=P B=F OR')
saveas(pf3or, 'c:\users\owner\desktop\graphs\ETP\A vs B vs Across III
A=P B=F vs within A=P B=F OR.jpg')
```

## *Bibliography*


1. Abiantun, R., & Savvides, M. (2009). Dynamic three-bin real AdaBoost using biased classification systems: An application in face detection. Paper presented at the *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on,* 1-6.
2. Asman, A. J., & Landman, B. A. (2011). Robust statistical label fusion through consensus level, labeler accuracy, and truth estimation (COLLATE). *Medical Imaging, IEEE Transactions on, 30*(10), 1779-1794.
3. Dawid, A. P. (1979). *Conditional independence in statistical theory*. *Journal of the Royal Statistical Society*. *Series B (Methodological), 41*(1), 1-31.
4. Gross, D. (1999). A neural network ATR for high range resolution radar signature recognition of moving ground targets. Paper presented at the *Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on, , 2* 1235-1239 vol.2.
5. Hill, J. M., Oxley, M. E., & Bauer, K. W. (2003). Receiver operating characteristic curves and fusion of multiple classification systems. Paper presented at the *Information Fusion, 2003. Proceedings of the Sixth International Conference of, , 2* 815-822.
6. Hong-Hee Won, & Sung-Bae Cho. (2003). Paired neural network with negatively correlated features for cancer classification in DNA gene expression profiles. Paper presented at the *Neural Networks, 2003. Proceedings of the International Joint Conference on, , 3* 1708-1713 vol.3.

7. Kennedy, H. L. (2012). *Fusion of possibly biased location estimates using gaussian mixture models*. *Information Fusion, 13*(3), 214-222.

8. Khreich, W., Granger, E., Miri, A., & Sabourin, R. (2012). *Adaptive ROC-based ensembles of HMMs applied to anomaly detection*. *Pattern Recognition, 45*(1), 208-230. doi: 10.1016/j.patcog.2011.06.014

9. Khreich, W., Granger, E., Miri, A., & Sabourin, R. (2010). *Iterative boolean combination of classification systems in ROC space: An application to anomaly detection with HMMs*. *Pattern Recognition, 43*(8), 2732-2752.

10. Klein, L. A. (1993). A boolean algebra approach to multiple sensor voting fusion. *Aerospace and Electronic Systems, IEEE Transactions on, 29*(2), 317-327.

11. Landman, B. A., Asman, A. J., Scoggins, A. G., Bogovic, J. A., Fangxu Xing, & Prince, J. L. (2012). Robust statistical fusion of image labels. *Medical Imaging, IEEE Transactions on, 31*(2), 512-522.

12. Leap, N. J., Clemans, P. P., Bauer, K. W., & Oxley, M. E. (2008). *An investigation of the effects of correlation and autocorrelation on classification system fusion and optimal classification system ensembles*. *International Journal of General Systems, 37*(4), 475-498. doi: 10.1080/030810701300211

13. Liggins, M. E.,II, Chee-Yee Chong, Kadar, I., Alford, M. G., Vannicola, V., & Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE, 85*(1), 95-107.

14. Luo, R. C., Chih Chia Chang, & Chun Chi Lai. (2011). Multisensor fusion and integration: Theories, applications, and its perspectives. *Sensors Journal, IEEE, 11*(12), 3122-3138.

15. Padgett, C., & Woodward, G. (1997). A hierarchical, automated target recognition algorithm for a parallel analog processor. Paper presented at the *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on,* 374-379.

16. Pang, Y. W., Deng, J. P., & Yuan, Y. (2012). *Incremental threshold learning for classification system selection*. *Neurocomputing*, *89*, 89-95. doi: 10.1016/j.newcom.2012.01.012

17. Parker, D. R., Gustafson, S. C., Oxley, M. E., & Ross, T. D. (2010). Development of a bayesian framework for determining uncertainty in receiver operating characteristic curve estimates. *Knowledge and Data Engineering, IEEE Transactions on, 22*(1), 31-45.

18. Petrakos, M., Kannelopoulos, I., Benediktsson, J. A., & Pesaresi, M. (2000). The effect of correlation on the accuracy of the combined classification system in decision level fusion. Paper presented at the *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International, , 6* 2623-2625 vol.6.

19. Pilcher, C., & Khotanzad, A. (2008). Statistical performance of classification systems for a maritime ATR task. Paper presented at the *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National,* 163-167.

20. Ralescu, A., & Visa, S. (2006). Fuzzy classification systems versus cost-based bayes classification systems. Paper presented at the *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual Meeting of the North American,* 302-305.

21. Sani, Y., Mohamedou, A., Ali, K., Farjamfar, A., Azman, M., & Shamsuddin, S. (2009). An overview of neural networks use in anomaly intrusion detection systems. Paper presented at the *Research and Development (SCOReD), 2009 IEEE Student Conference on,* 89-92.

22. Schubert, C. M. (September 2005). *Quantifying correlation and its effects on system performance in classification system fusion. PhD Dissertation, AFIT/DS/ENC/05-01 Air Force Institute of Technology.*

23. Schubert, C. M., Oxley, M. E., & Bauer, K. W., Jr. (2005). A comparison of ROC curves for label-fused within and across classification system systems. Paper presented at the *Information Fusion, 2005 8th International Conference on, , 1* 8 pp.

24. Schubert, C. M., Oxley, M. E., & Bauer, K. W., Jr. (2005). *Quantifying the performance of fused correlated multiple classification systems. Signal Processing, Sensor Fusion, and Target Recognition XIV Proc. of SPIE, 5809,* 390-401. doi: 10.1117/12.602956

25. Schubert, C. M., Thorsen, S. N., & Oxley, M. E. (2011). *The ROC manifold for classification systems. Pattern Recognition, 44*(2), 350-362. doi: 10.1016/j.patcog.2010.07.025

26. Sinha, N. K., Gupta, M. M., & Rao, D. H. (2000). Dynamic neural networks: An overview. Paper presented at the *Industrial Technology 2000. Proceedings of IEEE International Conference on, , 1* 491-496 vol.2.

27. Solberg, A. S., Storvik, G., & Fjortoft, R. (2002). A comparison of criteria for decision fusion and parameter estimation in statistical multisensor image

classification. Paper presented at the *Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International, , 1* 72-74 vol.1.

28. Swaving, M., VanHouwelingen, H., Ottes, F. P., & Steerneman, T. (1996). *Statistical comparison of ROC curves from multiple readers*. *Medical Decision Making, 16*(2), 143-152. doi: 10.1177/0272989X9601600206
29. Thompson, M. L. (2003). *Assessing the diagnostic accuracy of a sequence of tests*. *Biostatistics, 4*(3), 341-351.
30. Wangqiang Niu, Jin Zhu, Wei Gu, & Jianxin Chu. (2009). Four statistical approaches for multisensor data fusion under non-gaussian noise. Paper presented at the *Control, Automation and Systems Engineering, 2009. CASE 2009. IITA International Conference on,* 27-30.
31. Werbos, P. J. (1991). An overview of neural networks for control. *Control Systems, IEEE, 11*(1), 40-41.
32. Willett, P., Swaszek, P. F., & Blum, R. S. (2000). The good, bad and ugly: Distributed detection of a known signal in dependent gaussian noise. *Signal Processing, IEEE Transactions on, 48*(12), 3266-3279.
33. Zhang Xinhua. (1998). An information model and method of feature fusion. Paper presented at the *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on, , 2* 1389-1392 vol.2.

| REPORT DOCUMENTATION PAGE | | | | | | *Form Approved* *OMB No. 074-0188* |
|---|---|---|---|---|---|---|
| The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.** | | | | | | |
| **1. REPORT DATE** *(DD-MM-YYYY)* 21-08-2012 | | **2. REPORT TYPE** **Master's Thesis** | | | | **3. DATES COVERED** *(From – To)* March 2011 – September 2012 |
| **4. TITLE AND SUBTITLE** **Quantifying Performance Bias in Label Fusion** | | | | | | **5a. CONTRACT NUMBER** **DE AC05-06OR23100** |
| | | | | | | **5b. GRANT NUMBER** |
| | | | | | | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)** **Venzin, Alexander M.** | | | | | | **5d. PROJECT NUMBER** |
| | | | | | | **5e. TASK NUMBER** |
| | | | | | | **5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)** Air Force Institute of Technology Graduate School of Engineering and Management(AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765 | | | | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GAM/ENC/12-04 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** Mr. Mark M. Derriso Air Force Research Laboratory/RBSA 2130 8th Street WPAFB, OH 45433 (937) 255-8534 | | | | | | **10. SPONSOR/MONITOR'S ACRONYM(S)** **AFRL/RB** |
| | | | | | | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |
| **12. DISTRIBUTION/AVAILABILITY STATEMENT** **DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RESLEASE; DISTRIBUTION UNLIMITED.** | | | | | | |
| **13. SUPPLEMENTARY NOTES** **This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.** | | | | | | |
| **14. ABSTRACT** Classification systems are employed to remotely assess whether an element of interest falls into a "target" class or "non-target" class. These systems have uses in fields as far ranging as biostatistics to search engine keyword analysis. The performance of the system is often summarized as a trade-off between the proportions of elements correctly labeled as "target" plotted against the number of elements incorrectly labeled as "target." These are empirical estimates of the true positive and false positive rates. These rates are often plotted to create a receiver operating characteristic (ROC) curve that acts as a visual tool to assess classification system performance. The research contained in this thesis focuses on the label fusion technique and the bias that can occur when using incorrect assumptions regarding the partitioning of the event set. This partitioning may be defined in terms of what will be called *within* and *across* label fusion. The major goals of this work are the formulaic development and quantification of performance bias between different types of *across* and *within* label fusion and analysis of the effects of individual classification system performance, correlation, and target environment on the magnitude of bias between these two types of label fusion. | | | | | | |
| **15. SUBJECT TERMS** **Label fusion, classification, performance bias, simulation, receiver operating characteristic** | | | | | | |
| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** **Dr. Christine Schubert Kabban (ENC)** | |
| **REPORT** U | **ABSTRACT** U | **c. THIS PAGE** U | **SAR** | 296 | **19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 EXT:4549 christine.schubertkabban@afit.edu | |