Journal of Machine Learning Research 20 (2019) 1-40 Submitted 2/18; Revised 9/19; Published 11/19

# Unsupervised Evaluation and Weighted Aggregation of Ranked Classification Predictions

**Mehmet Eren Ahsen**[1,2] AHSEN@ILLINOIS.EDU
**Robert M Vogel**[2,3] R.VOGEL@IBM.COM
**Gustavo A Stolovitzky**[2,3] GUSTAVO@US.IBM.COM
[1] *University of Illinois at Urbana-Champaign*
*Department of Business Administration,*
*1206 S 6th St, Champaign, IL 61820, USA*
[2] *Icahn School of Medicine at Mount Sinai*
*Department of Genetics and Genomic Sciences*
*One Gustave Levy Place, Box 1498*
*New York, NY , USA*
[3] *IBM T.J. Watson Research Center*
*1101 Kitchawan Road, Route 134,*
*Yorktown Heights, N.Y., 10598, USA.*

**Editor:** Samy Bengio

## Abstract

Ensemble methods that aggregate predictions from a set of diverse base learners consistently outperform individual classifiers. Many such popular strategies have been developed in a supervised setting, where the sample labels have been provided to the ensemble algorithm. However, with the rising interest in unsupervised algorithms for machine learning and growing amounts of uncurated data, the reliance on labeled data precludes the application of ensemble algorithms to many real world problems. To this end we develop a new theoretical framework for ensemble learning, the Strategy for Unsupervised Multiple Method Aggregation (SUMMA), that estimates the performances of base classifiers and uses these estimates to form an ensemble classifier. SUMMA also generates an ensemble ranking of samples based on the confidence score it assigns to each sample. We illustrate the performance of SUMMA using a synthetic example as well as two real world problems.

**Keywords:** Ensemble learning, Ensemble classifier, Unsupervised Learning, AUC, Spectral Decomposition

## 1. Introduction

Algorithmic solutions to typical machine learning problems are quickly increasing in number and complexity. However, no algorithm performs best under every circumstance: there is no one-size-fits-all solution in machine learning. A mathematical formulation of this phenomena is known as the "no free lunch" theorem (Wolpert, 1996). On the other hand, it has long been appreciated that the combination of different algorithms, both in classification and regression tasks, results in more robust solutions (Dietterich, 2002).

Indeed, the endeavor of combining predictions or preferences from multiple sources has been institutionalized and incorporated in our everyday decision making. For example, in democratic elections the candidate who gets the most votes wins, or in online purchases the products with better customer reviews are more likely to be chosen. In social sciences this collaborative decision-making process is known as the Wisdom of Crowds (WOC) (Surowiecki, 2005). In the machine learning literature, the process of combining multiple base learners is known as ensemble learning.

Ensemble algorithms can be divided into two main categories, namely supervised and unsupervised ensemble algorithms. Supervised ensemble algorithms can be further subdivided into two main subcategories. The first represents homogeneous ensemble algorithms that consist of training several instances of a single type of algorithm on various splits of labeled data. This subcategory of methods generates diversity by subsampling training examples (bagging) (Breiman, 1996) or assigning weights to training examples (boosting) (Freund and Schapire, 1997; Schapire, 1990) utilizes a single type of base classifier to build the ensemble classifier. Homogeneous ensembles may suffer in applications where the data is very complex, and consequently the most appropriate base classifier model is not known *a priori*. The second subcategory is known as heterogeneous ensemble algorithms, where each base classifier represents a distinct algorithm. Two popular heterogeneous ensemble methods are a form of meta-learning called stacking (Wolpert, 1992) as well as the ensemble selection procedure proposed in (Caruana et al., 2004). Among them, stacking trains a higher-level classifier over the predictions of base classifiers, while ensemble selection uses an iterative strategy to select an optimal set of base classifiers that balances diversity and performance.

Although supervised heterogeneous ensembles often out-perform homogeneous ensembles (Niculescu-Mizil et al., 2009; Gashler et al., 2008), there are several limitations regarding their use in real life scenarios. Along with the risk of overfitting associated with the meta-training of ensemble classifiers, another very important limitation of heterogeneous supervised ensemble algorithms is that in many domains labeled data for training ensemble classifiers are scarce and, in some cases, there may not be any available at all. For such cases, unsupervised learning methods might be more appropriate. Unsupervised ensemble learning has been used successfully in diverse applications including computational biology (Marbach et al., 2012), crowdsourcing in natural language tasks (Snow et al., 2008), and business (Tsai and Hsiao, 2010). The simplest unsupervised ensemble algorithm creates an ensemble score by an unweighted average of the ranks assigned by the base classifiers to each item being classified (Marbach et al., 2012; Emerson, 2013). We will call this simple unsupervised ensemble method as the "Wisdom of Crowds" (WOC) approach (Surowiecki, 2005). The WOC approach suffers when the majority of base classifiers perform poorly. A strategy to mitigate this shortcoming would be to assign each base classifier a weight commensurate (e.g., proportional) to its performance. However, this approach may not be advisable or applicable when (1) the biases of the training sets are different from the test set, or (2) there is scarce or no prior labeled data sets to calculate performance.

An early example of ensemble classification is given in Dawid and Skene (1979), where the authors propose to infer the performance of each base classifier and the true class labels associated with each sample together. Specifically, they recast the problem so that a maximum likelihood solution could be estimated by the Expectation Maximization (EM) al-

gorithm (Dempster et al., 1977). While elegant, this solution to the unsupervised ensemble construction suffers from the known limitations of the EM algorithm for non-convex optimization problems. Another important limitation is the strong assumption of conditional independence which might be violated in practical settings.

In this work, we propose a new theoretical framework for unsupervised ensemble learning, the Strategy for Unsupervised Multiple Method Aggregation (SUMMA). Our work is a generalization of the Spectral Meta-Learner (SML) theory developed in Parisi et al. (2014). In SML, the authors starting point is the covariance matrix of predicted binary labels of a set of base classifiers. They show that under the assumption of conditional independence of the class predictions of base classifiers given the true class labels, the off-diagonal elements of the covariance matrix are related to the balanced accuracy of each base classifier. Many popular classification algorithms such as logistic regression (Harrell, 2001), SVM (Support Vector Machines) (Cortes and Vapnik, 1995), and deep learning algorithms (Chollet, 2017) produce continuous scores that can be interpreted as a measure of the confidence assigned by the base classifier that a given item is of one of the two binary classes. By forcing base classifiers to produce binary predictions SML is not only ignoring the readily available continuous scores outputted by typical base classifiers, but it also runs into the risk of overfitting by requiring each algorithm to define a threshold to binarize its output. In the case of SUMMA, our starting point is the covariance matrix of ranked predictions which can be created by rank ordering the continuous scores outputted by the base classifiers. We show that the off-diagonal entries of this covariance matrix are related to the Area Under the Receiver Operating Characteristics Curve (AUC) (Marzban, 2004) of the base classifiers. Under the assumption of conditional independence of the ranks assigned by base classifiers to an item given the class of the item, the SUMMA algorithm:

1. Infers the empirical performance, as measured by the AUC, of each base classifier without labeled data,

2. Uses the inferred performances of the base classifiers to generate an aggregate continuous score for each sample, and

3. Uses these scores to predict the binary class labels for each sample.

To exemplify our theoretical results we first apply SUMMA to a simulated dataset where our theoretical assumption of conditional independence strictly holds. We also apply SUMMA to two datasets taken from real applications where we show that SUMMA performs robustly even if the conditional independence assumption is slightly violated.

## 2. Problem Setup

Given a classification problem, we assume that an instance pair $(X, Y) \in \mathcal{X} \times \{0, 1\}$ is a random vector with probability density function $p(x, y)$ and marginals $P_X(x)$ and $P_Y(y)$, where the set $\mathcal{X}$ denotes the feature space, and without loss of generality, $y = 0$ denotes the negative class and $y = 1$ denotes the positive class. Let $\{g_i\}_{i=1}^{M}$ represent an ensemble of classifiers with unknown reliability, where each classifier, $g_i : \mathcal{X} \to \mathbb{R}$, is a mapping from the feature space $\mathcal{X}$ to real numbers. We assume that each classifier is trained on its own possibly labeled dataset which is unknown to us and produces confidence scores

$g_i(x_k)$ associated with a set of $N$ instances $\mathcal{D} = \{(x_k)\}_{k=1}^N \subset \mathcal{X}$, whose true but unknown class labels are denoted by the vector $\boldsymbol{y} = [y_1, \cdots, y_N]^T$. We let $N_1 := \sum_{i=1}^N y_i$ denote the number of positive instances, $N_0 = N - N_1$ denote the number of negative instances and $\rho = N_1/N$ denote the *prevalence* of the positive class, which we simply denote as prevalence in the text. We can interpret the output of each classifier as a measure of the relative confidence that the sample belongs to one of the classes, which, without loss of generality, will be assumed to be class 1. For example, in a Bayesian framework the output of a classifier is the posterior probability that a sample belongs to the positive class. In the case of SVM, the output is the distance to the separating hyperplane.

Recent research suggests that appropriate calibration of the classifier outputs will boost the performance of the ensemble classifier (Whalen and Pandey, 2013; Bella et al., 2013). In the current manuscript, we will use the rank transformation as a calibration tool due to its theoretical implications presented in the subsequent sections. Accordingly, let $\Omega = \{(x_1, y_1), \cdots, (x_N, y_N) \in (\mathcal{X} \times \{0,1\})^N : \sum_{i=1}^N y_i = N_1\}$ denote the space of all $N$ i.i.d realizations of $\mathcal{X} \times \{0,1\}$ with exactly $N_1$ positive instances. For each $T \in \Omega$ and classifier $i$, consider the following vector $g_i(T) = [g_i(x_1), \cdots, g_i(x_N)]^T \in \mathbb{R}^N$. By using the well-ordering property of the real numbers, we can map $g_i(T)$ to a rank vector $\mathbf{r} = [r_1, ..., r_N]^T \in \mathcal{S}_N$, where $\mathcal{S}_N$ denotes the set of all permutations of the integers $\{1, \cdots, N\}$. As in the recent literature (Agarwal et al., 2005), we assume that ties, i.e. $g_i(x_j) = g_i(x_k)$ for $j \neq k$, are broken uniformly at random. In this case, w.l.o.g., we assume that $g_i$ assigns ranks to tied samples according to their position in the vector $g_i(T)$. Throughout the rest of the manuscript, unless otherwise noted when we say we are given a realization $T \in \Omega$, we assume that we do not observe the labels associated with each sample and only observe the feature vector $x_k$ associated with each sample. With the above notation, we let $P(R_i = r|y_k)$ denote the probability, over all realizations from $\Omega$, that a sample of class $y_k \in \{0,1\}$ has been assigned rank $r \in \{1, \cdots, N\}$ by the classifier $i$. For simplicity, we will refer to this quantity as $P_i(r|y_k)$ and also let $P\left(R_1 = r_{1k}, R_2 = r_{2k}, \cdots, R_M = r_{Mk}|y_k\right)$ denote the joint probability of each classifier $i$ assigning rank $r_{ik}$ to a sample $k$ of class $y_k$.

## 3. Theory

### 3.1. Performance Metric

We start by defining the main performance metric that is primarily used throughout the paper.

**Definition 1** *The performance of the $i^{th}$ classifier is measured by,*

$$\Delta_i = \mathbb{E}[R_i|y = 0] - \mathbb{E}[R_i|y = 1],$$

*where $\mathbb{E}[R_i|y = j]$, for $j = 0, 1$, represents the average rank given the respective class for the $i^{th}$ classifier.*

$\Delta_i$ is defined over the set $\Omega$ and as such it may depend on $N$ and $\rho$. Note that a random method has $\Delta_i$=0. Intuitively, this is because random methods are those unable to rank samples according to the latent class, a consequence of the fact that for a random method $P_i(r|y = 1) = P_i(r|y = 0) = \mathcal{U}(1, N)$, where $\mathcal{U}(1, N)$ denotes the uniform distribution on the

set of integers $\{1, 2, ..., N\}$. On the other hand if $|\Delta_i| > 0$, then method $i$ is an informative method and can discriminate rank assignments by the sample class.

We will assume that all methods $i = 1, ..., M$ use the same convention consisting of assigning ranks to the examples predicted to be in the positive class in the lower range values of the interval $1, ..., N$, and the examples predicted to be in the negative class to the upper range of the same interval. In this way, a perfect classifier $g_P$ would assign the ranks $[1, 2, ..., N_1]$ to the $N_1$ positive examples, and the ranks $[N_1 + 1, ..., N]$ to the negative examples. In this case $\mathbb{E}[R_P|y = 1] = (N_1 + 1)/2$, $\mathbb{E}[R_P|y = 0] = (N + N_1 + 1)/2$ and $\Delta_P = N/2$.

### 3.2. Conditionally Independent Classifiers

In this section we first define the conditional independence assumption which is central to our theoretical results. We call the classifiers, $\{g_1, \cdots, g_M\}$, $l^{th}$ order conditionally independent, if for any $L = \{g_{i_1}, \cdots, g_{i_l}\} \subset \{g_1, \cdots, g_M\}$ their conditional distribution factorizes,

$$P\left(R_{i_1} = r_{i_1 k}, R_{i_2} = r_{i_2 k}, \cdots, R_{i_l} = r_{i_l k}|y_k\right) = P_{i_1}(r_{i_1 k}|y_k)P_{i_2}(r_{i_2 k}|y_k)\cdots P_{i_l}(r_{i_l k}|y_k). \quad (1)$$

Next we prove that under the assumption of conditionally independent classifiers, their higher-order covariance tensors have a specific form.

**Theorem 2** *Suppose the classifiers $\{g_1, \cdots, g_M\}$ are $l^{th}$ order conditionally independent so that equation* (1) *holds. For any $n \leq l \leq M$, let the $n^{th}$ order covariance tensor, $\Sigma_n$ , be defined as*

$$\Sigma_n(1, \cdots, n) := \mathbb{E}\big[\big(R_1 - \mathbb{E}[R_1]\big)\cdots\big(R_n - \mathbb{E}[R_n]\big)\big], \quad (2)$$

*where w.l.o.g. we denoted a given subset of classifiers of size n by the set $\{1, \cdots, n\}$. Then the following equality holds,*

$$\Sigma_n(1, \cdots, n) = C(\rho)(\rho^{n-1} - (\rho - 1)^{n-1})\prod_{j=1}^{n}\Delta_j, \quad (3)$$

*where $C(\rho) := \rho(1 - \rho)$ and $\rho = N_1/N$ denotes the prevalence of the positive class.*

**Proof** See Appendix B. ■

Theorem 2 shows that under the assumption of conditionally independent ensemble members, the $n^{th}$ central moment contains information of each method performance, $\Delta_i$. In addition, note the symmetry $\rho \to 1 - \rho$, and $\Delta \to -\Delta$ of the $n$-th central moment. This symmetry is expected because the classes 1 and 0 are assigned by convention. As mentioned above, we assume that all the base classifiers have adopted the same convention of assigning lower (higher) ranks to items predicted to be in the positive (negative) class. Therefore if the positive class were now called negative and vice versa, then $\rho$ would change to $1 - \rho$, $\Delta$ would change to $-\Delta$, and the formula for the $n$-th central moment would remain invariant to the change of convention.

### 3.3. Connections between Classifier Performance and Covariance Matrix

The theory of the current section depends on the assumption that the base classifiers are mutually or second order conditionally independent. Classifiers that are nearly mutually conditionally independent may arise, for example, from experts with different technical backgrounds, or from algorithms that are based on different design principles or independent sources of information. Note that similar independence assumptions appear also in other works considering a setting similar to ours (Dawid and Skene, 1979; Parisi et al., 2014). For completeness, we next state the second order conditional independence assumption.

**Assumption 1** *The ensemble of classifiers* $\{g_i\}_{i=1}^{M}$ *are second order conditionally independent. Therefore, for any* $i \neq j \in \{1, \cdots, M\}$ *and any sample* $k$*,*

$$P\left(R_i = r_{ik}, R_j = r_{jk} | y_k\right) = P_i(r_{ik}|y_k) P_j(r_{jk}|y_k). \tag{4}$$

We let $\Sigma_2$ represent the covariance matrix between base classifier, i.e.

$$\Sigma_2(i,j) = \mathbb{E}[(R_i - \mathbb{E}[R_i])(R_j - \mathbb{E}[R_j])], \tag{5}$$

for given classifiers $i$ and $j$. We next demonstrate that the covariance matrix has a special decomposition under assumption 1.

**Corollary 3** *The covariance matrix* $\Sigma_2$ *as defined in Equation* (5) *can be expressed as,*

$$\Sigma_2(i,j) = \begin{cases} \frac{N^2-1}{12} & \text{if } i = j \\ C(\rho)\Delta_i\Delta_j & \text{if } i \neq j \end{cases}$$

*if Assumption 1 holds.*

**Proof** Under Assumption 1 all the classifiers are mutually conditionally independent. Let $i \neq j$, then the result follows from Theorem 2 by letting $n = l = 2$. Next, let $i = j$ and recall that $\mathbf{r}$ is a vector in which each element represents the unique rank of each sample $k \in \{1, \ldots, N\}$. Therefore, the variance, $\Sigma_2(i,i)$, of $\mathbf{r}$ is that a uniform discrete distribution, $(N^2-1)/12$, which completes the proof of the theorem. ■

Note that the off-diagonal entry $\Sigma_2(i,j)$ contains information on the performances $\Delta_i$ and $\Delta_j$ of base classifiers $i$ and $j$. An intuitive understanding of this result can be grasped as follows. For an arbitrary pair of base classifiers, not necessarily conditionally independent, their covariance can be decomposed into intraclass and interclass covariance. The intraclass covariance is often a manifestation of the similarities between the methodologies used in the classifiers, or the fact that base classifiers were trained on similarly distributed data. The interclass covariance, on the other hand, represents the agreement of ranking samples based on each latent class; hence, if two algorithms perform well, their interclass covariance will reflect that performance. It follows that if two base classifiers are conditionally independent, the intraclass covariance will vanish, making the measured covariance exclusively attributable to interclass covariance which in turn reflects the performance of the base classifiers.

Inspection of Corollary 3 motivates a strategy for estimating each base predictor performance metric, $\Delta_i$, from the covariance matrix. Here we see that $\Sigma_2$ can be decomposed as

$$\Sigma_2 = R - diag(R) + \frac{N^2 - 1}{12} I, \tag{6}$$

where $I$ is the identity matrix and $R := \lambda_c v v^T$ is a rank-one matrix, with

$$\lambda_c := C(\rho) \left\| \Delta \right\|_2^2, \tag{7}$$

where $\|\Delta\|_2$ denotes the euclidean norm of the vector $\Delta := [\Delta_1, \cdots, \Delta_M]^T \in \mathbb{R}^M$, and $v$ is the unit norm vector with entries defined as

$$v_i := \frac{\Delta_i}{\sqrt{\sum_{j=1}^{M} \Delta_j^2}}. \tag{8}$$

Note that $\lambda_c$ is dependent on $\rho$ which is unknown to us in our unsupervised setup. However, as we will see in the next section, SUMMA only needs an estimate of $v_i$, which is a quantity proportional to the estimated performance $\Delta_i$ of each base classifier $i$, to calculate the ensemble classifier. However, estimation of $\rho$ is necessary to estimate $\Delta_i$ and from it the AUC of base classifier $i$ as will be shown in Theorem 4. In Section 6, we present a way to estimate $\rho$.

It follows from Corollary 3 and the assumption that at least 3 base classifiers are better than random that $3 \leq M$ is a sufficient condition for estimating the quantities $v_i$'s. To see why we need the requirement of $3 \leq M$, observe that we have $(M(M-1)/2)$ known off-diagonal elements of $\Sigma_2$ but $M$ unknowns ($v_i$ for $i = 1, \cdots, M$), so if $M < 3$ the system is underdetermined. Moreover, note that if only a single classifier $i$ performs better than random, and the rest perform randomly, i.e.

$$v_i > 0, \quad v_j = 0 \quad \text{for } j \neq i,$$

then the off-diagonal entries of $\Sigma_2$ are exactly equal to zero in which case it is impossible to infer $v_i$. Similarly, if only two classifiers $v_i > 0$ and $v_j > 0$ are performing better than random, then we have again two unknowns but only one known non-zero off-diagonal element. Therefore, throughout the paper, we assume that at least three classifiers perform better than random. Under this assumption, the off-diagonal elements of the covariance matrix are equal to that of the rank-one matrix $R$ whose unique eigenvector is proportional to the vector of individual classifier performances.

Although $\Delta$ is both an intuitive and statistically principled measure of classifier performance, it is the expected performance metric over all the realizations in our probability space $\Omega$ and it can not be calculated explicitly as the probability distribution $P_i(r|y)$ is not readily available. For our results to be applicable in practice, we will use the unbiased empirical estimator of $\Delta_i$ defined as

$$\widehat{\Delta_i} = \frac{1}{N_0} \sum_{\{k|y_k=0\}} r_{ik} - \frac{1}{N_1} \sum_{\{k|y_k=1\}} r_{ik} = \widehat{R_{i|y=0}} - \widehat{R_{i|y=1}}, \tag{9}$$

where $T \in \Omega$ is a single realization from the probability space $\Omega$, $r_{ik}$ is the empirical rank assigned to sample $k$ by classifier $i$ by ordering the confidence scores $\{g_i(x_i)\}_{i=1}^N$ in descending order, and $\widehat{R_{i|y}}$ denotes the empirical conditional sample mean of rank assignment. We finish this section by showing the relation of $\widehat{\Delta_i}$ to another canonical empirical measure of performance for base classifier $i$, namely its AUC (Marzban, 2004).

**Theorem 4** *Given a ranked list of predictions and the corresponding sample class, $\{(r_{ik}, y_k)\}_{k=1}^N$, where $r_{ik}$ is the rank assigned to the sample $k$ by classifier $i$ and $y_k$ is the true class of sample $k$, we have the following equivalence*

$$AUC_i = \frac{\widehat{\Delta_i}}{N} + \frac{1}{2}, \tag{10}$$

*where $AUC_i$ is estimated using the rectangle rule.*

**Proof** For the definition of AUC and the proof see Appendix C. ■

It is easy to compute the empirical performance $\widehat{\Delta}$ using (9) if the labels associated with each sample are available. However, in our unsupervised setting, we only observe $r_{ik}$ which is the rank assigned to sample $k$ by classifier $i$. Indeed, one of the main contributions of the current manuscript is to infer $\widehat{\Delta}$ without access to class labels. For that purpose, we will use the relation between the covariance matrix $\Sigma_2$ and $\Delta$ derived in Section 3.2. The next section proposes two algorithms for this inference task.

## 4. Methods for Estimating Performances from the Covariance Matrix

As we stated in the previous section, due to the non-availability of $\Sigma_2$, we use $\widehat{\Sigma_2}$, the empirical covariance matrix, which is an unbiased estimator of the covariance matrix $\Sigma_2$, to infer the performance of base classifiers. In order to calculate $\widehat{\Sigma_2}$, we only require the rank predictions by each classifier. We then calculate an estimate $\widehat{v_i}$ of $v_i$ from $\widehat{\Sigma_2}$. For the sake of completeness, we will next define the empirical covariance matrix, $\widehat{\Sigma_2}$. As before, let $T \in \Omega$ be given, and an ensemble of classifiers $\{g_i\}_{i=1}^M$ ranks samples $\{x_k\}_{k=1}^N$ using $g_i(x_k)$ and assigns rank $r_{ik}$ to sample $k$. Then, the empirical covariance matrix $\widehat{\Sigma_2}$ is given by

$$\widehat{\Sigma_2}(i,j) = \frac{1}{N}\sum_{k=1}^{N}\left(r_{ik} - \frac{N+1}{2}\right)\left(r_{jk} - \frac{N+1}{2}\right).$$

In the next two subsections, we briefly describe two methods to calculate an estimator $\widehat{v_i}$ of $v_i$ from $\widehat{\Sigma_2}$. Our first approach formulates this task as a semi-definite program (SDP) which then can be solved using any SDP solver. Our second strategy is coming from the recent literature on the famous matrix completion problem, see e.g. (Candès and Recht, 2009; Barber and Ha, 2018; Cai et al., 2010; Jain et al., 2010). It is an iterative method that only requires calculating the largest singular value and associated singular vector at each iteration.

### 4.1. Semi-Definite Programming

In this section we present an SDP approach to estimate $\widehat{v_i}$ of $v_i$ from $\widehat{\Sigma_2}$. Here we will only describe the main idea and leave the details to the Appendix A. Let's assume that the covariance matrix $\Sigma_2$ is known and using the notation of equation (6), we will write it as $\Sigma_2 = R + D_{\Sigma_2}$, where $D_{\Sigma_2} := -diag(R) + \frac{N^2-1}{12}I$ is a diagonal matrix. Therefore, for $D = D_{\Sigma_2}$ the matrix $R$ is a feasible point for the following optimization problem,

$$\min_D rank(\Sigma_2 - D) \quad \text{s.t.} \quad D(i,j) = 0 \quad \text{for} \quad i \neq j, \quad \text{and} \quad \Sigma_2 - D \succeq 0, \tag{11}$$

where the positive semi-definite requirement on $\Sigma_2 - D$ is due to the non-negativity of the eigenvalues of the intended target $R = \lambda_c v v^T$ with $\lambda_c$ specified in Equation (7). We show in Lemma 10 (see Appendix A) that $R$ is the unique solution to the problem (11).

The optimization problem described in (11) is intuitive, but it is difficult to solve in practice. This is because the rank function is not convex (Candès and Recht, 2009). Indeed, the rank minimization problem for an arbitrary matrix is NP-hard (Jain et al., 2010). To circumvent this shortcoming, we chose to optimize the convex relaxation of the rank function, namely the nuclear norm, as is done in the matrix completion literature (Candès and Tao, 2010). For a given matrix $A$ the nuclear norm is defined as,

$$\|A\|_* = \sum_{i=1}^{M} \sigma_i(A), \tag{12}$$

where $\sigma_i(A)$ represents the $i^{th}$ singular value of $A$. Accordingly, the optimization problem in (11) can be relaxed to the following SDP,

$$\min_D \quad \|\Sigma_2 - D\|_* \quad \text{s.t.} \quad D \text{ is diagonal, and} \quad \Sigma_2 - D \succeq 0. \tag{13}$$

SDP is an active area of research, with applications in control theory and signal processing (Vandenberghe and Boyd, 1999), and many efficient solvers are readily available. Next we characterize the solutions of the semidefinite program in Equation (13).

**Theorem 5** *Let $\Sigma_2$ be the covariance matrix, then the optimization problem in Equation (13) has the unique solution $D_{\Sigma_2}$, provided that*

$$v_i^2 < \sum_{j \neq i} v_j^2, \quad \forall i \in \{1, 2, \ldots, M\}. \tag{14}$$

**Proof** Let $Q = \Sigma_2$ in Theorem 11. ■

The inference of the vector $v$ from the covariance matrix $\Sigma_2$ using Theorem 5 is unique up to its sign. Without further assumptions it is impossible to determine the sign of each coordinate of $v$, and hence the performance of each base classifier. In this manuscript, we assume that all base classifiers have adopted the right convention that lower (respect. higher) ranks correspond to samples predicted to be in the positive (respect. negative) class and that the majority of them perform better than random. As a result $v$ should have more positive entries than negative entries. Consequently, we solve the ambiguity between $v$ and $-v$ based upon which of the two choices has the greatest number of positive entries.

Up to this point we have assumed that the covariance matrix $\Sigma_2$ was known. However, in practice we only observe the sample covariance matrix $\widehat{\Sigma_2}$. To obtain the estimator $\widehat{R}$ we solve problem (13) using $\widehat{\Sigma_2}$, and interpret the vector associated with the leading singular value as the estimator $\widehat{v}$ such that $\widehat{R} = \widehat{\sigma_1}\widehat{v}\widehat{v}^T$.

### 4.2. An Iterative Approach

The iterative algorithm we present in this section and its convergence are inspired from the matrix completion literature and more details can be found in the recent papers (Barber and Ha, 2018; Jain et al., 2010). Similar to the previous section, we first assume that $\Sigma_2$ is known to us. Before presenting the iterative algorithm, let us define the set $\omega = \{(i,j) \,:\, i \neq j \in \{1, \cdots, M\}\}$. Then, using the notation of Barber and Ha (2018), we can formulate the following matrix completion problem:

$$\min_X \frac{||P_\omega(X) - P_\omega(\Sigma_2)||_F^2}{2} \quad s.t. \quad X \in C = \{X \in R^{M\times M} : rank(X) = 1, X = X^T, X \succeq 0\}, \tag{15}$$

where $||A||_F = \sqrt{\mathrm{Tr}(AA^T)}$ is the Frobenius norm of matrix $A$ and the linear operator $P_\omega(X)$ is defined as

$$P_\omega(X) = \begin{cases} X_{ij} & \text{if } (i,j) \in \omega \\ 0 & \text{if } (i,j) \notin \omega. \end{cases} \tag{16}$$

Again, it is obvious that $R = \Sigma_2 - D_{\Sigma_2} \in C$ and $||P_\omega(R) - P_\omega(\Sigma_2)||_F^2 = 0$, as such $R$ is a minimizer of the optimization problem in (15). Moreover, under the assumptions of Lemma 10, $R$ is the unique minimizer. Therefore, by solving the optimization problem in (15), we can recover the unknown vector $v$. The optimization problem in (15) is exactly the matrix completion problem presented in Barber and Ha (2018), with the additional fact that the matrix $R$ is of rank-one. From Theorem 3 of Barber and Ha (2018), under the assumption that $R$ has more than one element different from zero, which is the same as the condition $v_1 > 0, v_i = 0\ \forall i > 1$, the following gradient descent algorithm converges to $R$:

$$\begin{aligned} Y_t &= X_t - \nabla\left(\frac{1}{2}||P_\omega(X_t) - P_\omega(\Sigma_2)||_F^2\right) \\ X_{t+1} &= P_C(Y_t),, \end{aligned} \tag{17}$$

where $P_C(Y_t)$ is the projection of $Y_t$ into the set $C$. First note that

$$\nabla\left(\frac{1}{2}||P_\omega(X_t) - P_\omega(\Sigma_2)||_F^2\right) = P_\omega(X_t) - P_\omega(\Sigma_2),$$

and from Eckart - Young - Mirsky Theorem (Eckart and Young, 1936), we have

$$P_C(Y_t) = y_1(Y_t)u_1u_1^T,$$

where $y_1(Y_t)$ is the largest singular value of $Y_t$ and $u_1$ is the corresponding singular vector.

Since we do not observe $\Sigma_2$ in practice, we will use the empirical covariance matrix $\widehat{\Sigma_2}$ with the gradient descent algorithm given in (17) to calculate $\widehat{v}$ which is an estimator of $v$. The pseudo-code of the iterative algorithm is given in Algorithm 1.

**Algorithm 1** Find rank 1 matrix from off-diagonal observations of covariance matrix

Given $\widehat{\Sigma_2}$, fix $\epsilon > 0$, $t = 1$ and let $\lambda(0) = u(0) = 0$, $\lambda(1), u(1) \leftarrow SVD(\widehat{\Sigma_2})$,
where $\lambda(1)$, $u(1)$ represent the largest singular value and corresponding singular vector of $\widehat{\Sigma_2}$.
**while** $|\lambda(t) - \lambda(t-1))| > \epsilon$ **do**
    $Y(t) \leftarrow \widehat{\Sigma_2} - diag(\widehat{\Sigma_2}) + diag(\lambda(t)u(t)u(t)^T)$
    $t = t + 1$
    $\lambda(t), u(t) \leftarrow SVD(Y(t))$
**return** $[\ \widehat{\lambda}_c = \lambda(t), \widehat{v} = u(t)]$

## 5. Strategy for Unsupervised Multiple Method Aggregation: SUMMA

In the previous section, we showed how to estimate $v_i$, a quantity proportional to the performance of each base classifier, without knowing the labels associated with each sample. In this section, we develop a meta-learner that infers each sample's latent class and produces an aggregate ranking of samples using a weighted sum of base classifier rank predictions. As in the seminal work of Dawid and Skene (1979) and subsequent work by others (Nitzan and Paroush, 1982; Parisi et al., 2014), we cast the class inference task as a maximum likelihood estimation (MLE) problem.

As in the previous section, assume that a realization $T = \{(x_1, y_1), \cdots, (x_N, y_N)\} \in \Omega$ is given where the labels associated with samples are not available to us. The task at hand is to find the most likely class label, $y_k$, associated with each sample $k$, when the only available data are the rank predictions by conditionally independent base classifiers. Formally, the maximum likelihood estimate of $y_k$ is:

$$\widehat{y_k}^{\text{MLE}} = \underset{y}{\text{argmax}} \left\{ \sum_{i=1}^{M} \log \left( P_i \left( r_{ik} | y \right) \right) \right\}.$$

By application of Bayes' Theorem, we can write $P_i \left( r_{ik} | y \right) = P_i \left( y | r_{ik} \right) P(r_{ik}) / P(y)$. Using that the prior probability for the ranks is $P(r_{ik}) = 1/N$ and the prior probability for being in the positive class is the class prevalence ($\rho$ for class $y = 1$ and $1 - \rho$ for class $y = 0$), the MLE can be equivalently written as,

$$\widehat{y_k}^{\text{MLE}} = \Theta \left\{ M \log \left( \frac{1-\rho}{\rho} \right) + \sum_{i=1}^{M} \log \left( \frac{P_i \left( y_k = 1 | r_{ik} \right)}{1 - P_i \left( y_k = 1 | r_{ik} \right)} \right) \right\}. \tag{18}$$

The central challenge in applying the MLE is that the functional form of the probability distribution $P(y = 1|r)$ is *a priori* unknown. We are then faced with two choices: 1) *a priori* assume the functional form of the probability distribution $P(y = 1|r)$, or 2) infer a distribution. If we were to *a priori* assume a distribution, we would in effect be biasing the MLE and the dependence of the ensemble algorithm on the base classifier performance $\Delta$. This is problematic, because we have no reason to pick one distribution over another. A more principled approach would be to infer a distribution using information that is available to us and no more. The maximum entropy methodology for inferring distributions performs such a task (Jaynes, 1957). In short, by choosing the maximum entropy distribution we

are selecting the distribution which reproduces known statistical quantities and is otherwise maximally noncommittal to the remaining unknown moments. In our case we will assume that we have inferred the value of $\rho = N_1/N$ and the difference of conditional means $\Delta$, which will serve as constraints for the maximum entropy calculation. In the proceeding Lemma, we derive the maximum entropy distribution for $P(y = 1|r)$ given those constraints.

**Lemma 6** *The functional form of the maximum entropy probability distribution of the latent class label given the rank for a sufficiently weakly predictive classifier is approximated as*

$$P_i(y_k = 1|r_{ik}) = \left(1 + e^{\frac{12\Delta_i}{N^2-1}\left(r_{ik} - \frac{N+1}{2}\right) + \log\left(\frac{N-N_1}{N_1}\right)}\right)^{-1}.$$

**Proof** For the derivation see Appendix D. ■

Although we proved the above lemma for weakly predictive classifiers, we observed empirically through extensive simulations that the above formula is still a good approximation beyond that limit. Next we apply the maximum entropy probability distribution of Lemma 6 to derive a maximum likelihood estimator of each sample's latent class label.

**Theorem 7** *The maximum likelihood estimator (MLE) of sample $k$ is given as,*

$$\widehat{y_k}^{MLE} := \Theta\left\{\sum_{i=1}^{M} v_i \left(\frac{N+1}{2} - r_{ik}\right)\right\}, \tag{19}$$

*with $\Theta$ representing the unit step function.*

**Proof** Applying Lemma 6 to our maximum likelihood estimator in Equation (18),

$$\begin{aligned}
\widehat{y_k}^{\text{MLE}} &= \Theta\left\{M \log\left(\frac{1-\rho}{\rho}\right) - \sum_{i=1}^{M}\left[\frac{12\Delta_i}{N^2-1}\left(r_{ik} - \frac{N+1}{2}\right) + \log\left(\frac{N-N_1}{N_1}\right)\right]\right\} \\
&= \Theta\left\{\frac{12}{N^2-1}\sum_{i=1}^{M}\Delta_i\left(\frac{N+1}{2} - r_{ik}\right)\right\}
\end{aligned}$$

and recall from Equations (7, 8) that $\Delta_i = \sqrt{\frac{\lambda_c}{\rho(1-\rho)}}v_i$. Then by substitution,

$$y_k^{\text{MLE}} = \Theta\left\{\frac{12}{N^2-1}\sqrt{\frac{\lambda_c}{\rho(1-\rho)}}\sum_{i=1}^{M} v_i\left(\frac{N+1}{2} - r_{ik}\right)\right\} \tag{20}$$

where the terms preceding the sum have no influence to the image of the argument under the unit step function, and consequently may be ignored, which completes the proof. ■

Note that by our setup $\mathbb{E}[R_i] = (N+1)/2$, and also the rank assignment $r_{ik}$ are known. Therefore, in order to calculate the maximum likelihood estimate associated with each sample $k$, we only need to estimate $v_i$. For that we will use the empirical estimate $\hat{v}_i$

calculated by one of the two methods we proposed in Section 4. Then, the SUMMA estimate of $y_k$ is given as

$$\widehat{y_k}^{SUMMA} := \Theta\left\{\sum_{i=1}^{M} \hat{v}_i \left(\frac{N+1}{2} - r_{ik}\right)\right\}.$$

As the quantity $\sum_{i=1}^{M} \hat{v}_i \frac{N+1}{2}$ is the same for each sample $k$, the samples can be ranked using the SUMMA score, defined as

$$c_k^{SUMMA} = -\sum_{i=1}^{M} \hat{v}_i r_{ik}, \tag{21}$$

which can be interpreted as the score assigned to sample $k$ by SUMMA. The SUMMA score defined above is closely related to a more popular quantity in rank aggregation literature, namely the Borda count (Nitzan and Rubinstein, 1981). Borda count is a very popular aggregation technique in social choice theory (Sen, 1986) as well as computer science especially in the field of information retrieval (Subbian and Melville, 2011). Borda aggregation is equivalent up to the sign to ranking samples by assigning each ranker equal weight, i.e. letting $\widehat{v}_i = 1/M$ in (21). We would like to caution that in the pure rank aggregation problem the task in hand is to rank samples based on the preference of the ranker and there is usually a true but unknown ranking of samples that we want to estimate. However, in the information retrieval literature as well as the SUMMA, we rank the samples according to their probability belonging to the positive class and the aim is to find the true but unknown class label associated with each sample. Although in social endeavors such as democratic elections it is preferred to assign equal weight to each ranker, in machine learning it is desirable to assign each ranker a weight proportional to its performance. In the information retrieval literature, this weighted ranking scheme is called the weighted Borda-count (Aslam and Montague, 2001), where it is shown superior performance over Borda-count as well as other popular rank aggregation methods. However, in the literature, available algorithms for assigning weights in ranking problems require labeled data (supervision) (Aslam and Montague, 2001; Liu et al., 2010), or at least some semi-supervision (Balsubramani and Freund, 2015) or some prior information about each classifier's performance such as the public leaderboard scores in Kaggle (Sun et al.). The SUMMA rank aggregation is a weighted Borda-counting where the weights of each ranker, which are proportional to their classification performance, are estimated using unlabeled data. Therefore, the SUMMA score is closely related to rank aggregation literature and produces a robust ranking of samples in an unsupervised setting.

Apart from re-ranking a given set of samples and ranking classifiers based on their performance using a given matrix of ranked predictions, SUMMA can also use this matrix to learn an ensemble classifier to further classify an unseen sample. For this assume that we are given a dataset $T \in \Omega$ and an ensemble of classifiers $\{g_i\}_{i=1}^{M}$. Actually, in this situation the dataset $T$ will serve as an unlabeled training set to train an ensemble classifier. Suppose now we are given a new instance $(\psi, y_\psi) \in X \times \{0, 1\}$, where we want to get an estimate of $y_\psi$. We can achieve this by running SUMMA on the combined dataset of $N+1$ samples. As in the supervised setting, the quality and quantity of the samples in the training set would greatly affect the performance of the ensemble classifier.

## 6. Estimation of AUC of the Base Classifiers using the Third Order Covariance Tensor

The results of Section 4 help us estimate the vector $v$ whose entries are proportional to the performance of individual classifiers. Moreover, from (21) we see that this estimate, $\widehat{v}$ was sufficient to form the SUMMA ensemble classifier. However, it is not possible to estimate the actual performance, i.e. the AUC for each classifier, from the covariance matrix without *a priori* knowledge of the sample class prevalences. To address this shortcoming, we develop a strategy for estimating the prevalence from the third order covariance tensor of unlabeled rank data. Our approach is similar to that of Jaffe et al. (2015); however, we have extended the iterative algorithm presented in Section 4 by using the generalization of singular value decomposition to tensor decomposition, (Karami et al., 2012). Given three classifiers $i, j$ and $k$, the third order covariance tensor is defined as

$$\Sigma_3(i,j,k) = \mathbb{E}\left[\Big(R_i - \mathbb{E}[R_i]\Big)\Big(R_j - \mathbb{E}[R_j]\Big)\Big(R_k - \mathbb{E}[R_k]\Big)\right]. \tag{22}$$

Previously, we assumed that the rank predictions by base classifiers were mutually conditionally independent. For the theory in this section, we will extend this to triplets and will make the following assumption.

**Assumption 2** *The ensemble of classifiers $\{g_i\}_{i=1}^M$ are third order conditionally independent. Therefore, for any $i \neq j \neq l \in \{1, \cdots, M\}$ and sample $k$,*

$$P\left(R_i = r_{ik}, R_j = r_{jk}, R_l(x_k) = r_{lk} | y_k\right) = P_i(r_{ik}|y_k) P_j(r_{jk}|y_k) P_l(r_{lk}|y_k), \tag{23}$$

*for any $y_k \in \{0, 1\}$.*

Using this observation we present the following corollary of Theorem 2.

**Corollary 8** *Under Assumption 2, for any $i \neq j \neq l$ the covariance tensor is given as*

$$\Sigma_3(i,j,l) = C(\rho)(2\rho - 1)\Delta_i \Delta_j \Delta_l. \tag{24}$$

**Proof** Let $i$, $j$ and $l$ be integers in the set $\{1, 2, \ldots, M\}$ such that $i \neq j \neq l$. Then the result trivially follows from Theorem 2 with $n = 3$. ■

Corollary 8 shows that the covariance tensor $\Sigma_3$ is off-diagonal rank-one given by $\Sigma_3 \approx \lambda_t v \otimes v \otimes v$, where $\lambda_t := \Big((C(\rho)(2\rho-1)\Big)||\Delta||_2^3$. Recall from Equation (7), $\lambda_c = C(\rho)\left\|\Delta\right\|_2^2$. Next, let $\beta := \lambda_t^2/\lambda_c^3$ and observe that

$$\begin{aligned} \beta \quad &= \quad (2\rho-1)^2/(C(\rho)) \\ &\implies \quad \rho^2(\beta+4) - (\beta+4)\rho + 1 = 0. \end{aligned} \tag{25}$$

Therefore we can explicitly solve for $\rho$

$$\rho = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{\beta}{\beta+4}} \tag{26}$$

where we use the + sign if $\lambda_t$ is negative and the $-$ sign otherwise. By rearranging the terms in Equation (25) we find that

$$C(\rho) = \frac{1}{\beta + 4} \implies \|\Delta\| = \sqrt{\lambda_c(\beta + 4)}, \tag{27}$$

which gives us the norm of $\Delta$.

Again since $\Sigma_3$ is not available in practical situations, we will use the empirical covariance tensor $\widehat{\Sigma}_3$ defined as

$$\widehat{\Sigma}_3(i,j,l) = \frac{1}{N}\sum_{k=1}^{N}\left(r_{ik} - \frac{N+1}{2}\right)\left(r_{jk} - \frac{N+1}{2}\right)\left(r_{lk} - \frac{N+1}{2}\right).$$

In order to find an estimator of $\lambda_t$ from the third order statistics, we extend our iterative algorithm for decomposing the covariance matrix by using tensor SVD (tSVD) ((Kolda and Bader, 2009)). Pseudo-code for the generalized iterative algorithm is given in Algorithm 2.

**Algorithm 2** Find rank 1 matrix from off-diagonal observations of covariance tensor

Given $\widehat{\Sigma_3}$, fix $\epsilon > 0$, $t = 1$ and let $\lambda(0) = u(0) = 0$, $\lambda(1), u(1) \leftarrow \text{tSVD}(\widehat{\Sigma_3})$,
where $\lambda(1)$, $u(1)$ represent the largest singular value and corresponding singular tensor of $\widehat{\Sigma_3}$.
**while** $|\lambda(t) - \lambda(t-1)| > \epsilon$ **do**
    $Y(t) = \widehat{\Sigma}_3 - diag(\widehat{\Sigma}_3) + diag(\lambda(t)u(t)u(t)^T)$
    t=t+1
    $\lambda(t), u(t) \leftarrow \text{tSVD}(Y(t))$
**return** $[\widehat{\lambda}_t = y_1(X_t), \widehat{v} = u_1]$

Algorithm 2 gives us an estimate of the singular value of the covariance tensor $\lambda_t$ which is denoted as $\widehat{\lambda}_t$. Note that we can estimate both $\lambda_c$ and $\lambda_t$ using the iterative algorithms Algorithm 1 and Algorithm 2, respectively. This allows us to calculate an estimate of $\beta$ which in turn gives us an estimate of $\|\Delta\|_2$ from (27). Knowledge of $\|\Delta\|$ allows us to estimate the prevalence of each class label and for each $i$ base classifiers to compute an estimate of $\Delta_i$ using equation (7). Furthermore, we can use this estimate of $\Delta_i$ to compute the AUC for each $i$ classifier using the closed form formula given in Theorem 4. Apart from the iterative approach we proposed, other methods for estimating $\rho$ based on a restricted likelihood approach (Jaffe et al., 2015) or on spectral decomposition (Jain and Oh, 2014) exists albeit with additional assumption such as higher order conditional independence.

## 7. Examples of Applications of SUMMA

In this section we apply the SUMMA methodology and assess its performance in different example settings, including i) synthetic data, ii) predictions submitted to a crowd-sourced challenge and iii) several classification problems in different domains using datasets available from the UCI Machine Learning Repository (Lichman 2013). For the computations in this section we have used the R language on a personal laptop which has 4 computational cores and 16GB of RAM. Running the SUMMA algorithm is very fast (less than 5 minutes in

the worst case); and apart from the calculation of the covariance matrix and tensor, it is unaffected by the number of samples.

In the instances in which classifiers return the same score for different samples, their corresponding ranks will be tied. In those cases we will break the ties by assigning those samples random but unique ranks ranging between the immediately higher and lower untied ranks.

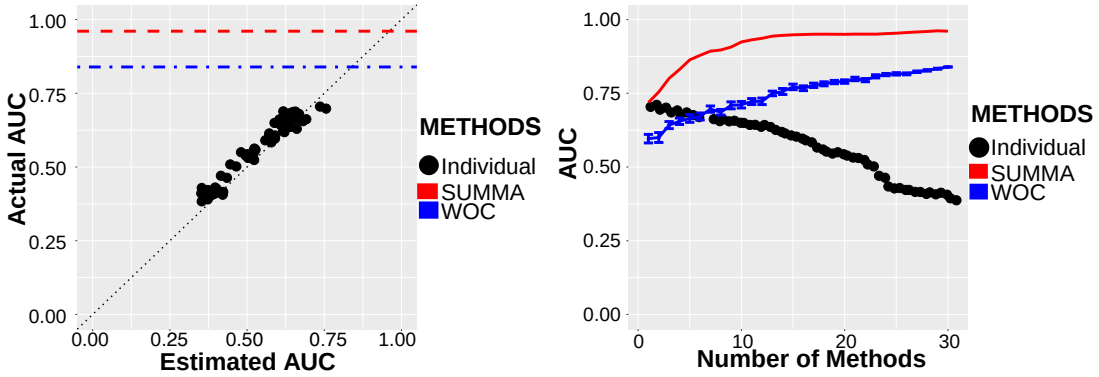
### 7.1. Synthetic Data Examples

In this section we use synthetic data to illustrate the ability of SUMMA to infer the AUC of individual methods as well as the performance improvements obtained using the SUMMA ensemble. We investigate the influence of the number of methods, the number of samples, and class prevalence on SUMMA performance. Each data set represents $N$ sample rank predictions from $M$ conditionally independent base classifiers. We generated synthetic predictions by producing random scores from two Gaussian distributions, each of which is used to simulate scores from the negative and positive class. The AUC of each base classifier was controlled by adjusting the parameters (mean and variance) of the respective class specific Gaussian distributions using the closed form formula of AUC for normally distributed class specific scores (Marzban, 2004):

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

where $\mu_i$ and $\sigma_i^2$ denote the mean and variance for the positive ($i = +$) and negative ($i = -$) classes, respectively, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. The conditional independence assumption between classifiers was satisfied by independently sampling from the Gaussian distributions associated with each base classifier. Once the samples were generated, we converted the scores to sample ranks using the convention that lower (higher) ranks correspond to samples predicted to be in the positive (negative) class.

For the first part of the analysis, we generated synthetic predictions for $M = 30$ base classifiers, with a total of $N = 1000$ samples of which 500 belongs to the positive class. We adjusted the parameters of the Gaussian distributions such that the distribution of base classifier AUC values was uniformly distributed between $(0.4, 0.74)$. Under this setting, Figure 1a shows that SUMMA reliably estimated the AUC of each base classifier with a correlation coefficient of 0.95 between the estimated and actual AUC. In addition, Figure 1a shows that the SUMMA ensemble (red, AUC=0.95) out-performs the best individual classifier (AUC=0.74), and the WOC ensemble (blue, AUC=0.89), which aggregates predictions by averaging the sample ranks of the base classifiers. Next we investigate the effect of changing the number of base classifiers. Figure 1b shows how the performances of the SUMMA and WOC ensembles change with the addition of extra classifiers. Here, we estimated the performance of base classifiers using the covariance of all 30 classifiers but constructed the $n^{th}$ SUMMA ensemble by aggregating the top $n$ performing classifiers where we rank classifiers based on their performance as estimated by SUMMA. Unlike SUMMA, in this unsupervised setting the WOC ensemble has no knowledge of the performances of each base classifier. Hence, for the WOC ensemble we randomly sub-selected $n$ methods

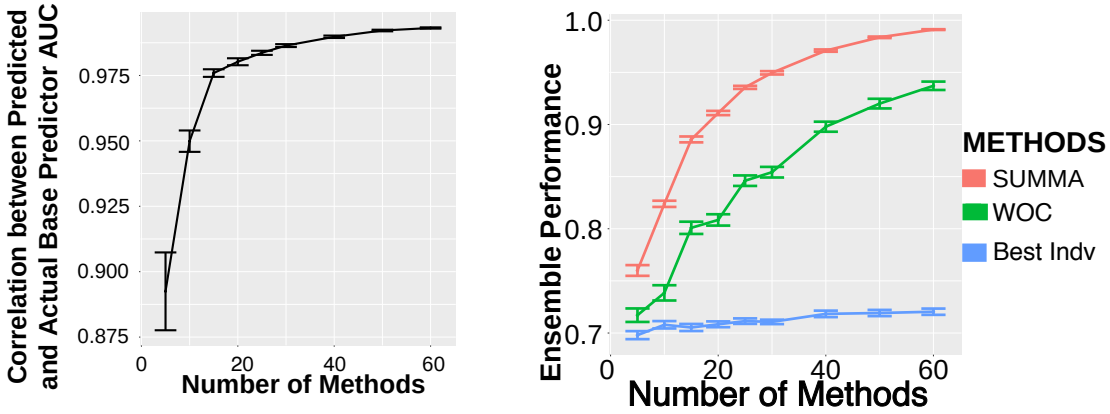



(a) Comparison between the SUMMA inferred AUC of the base classifiers, and their actual AUC

(b) AUC of the SUMMA ensemble as a function of the number of base classifiers

Figure 1: Validation and analysis of the SUMMA results with simulated data for $N = 1000$ samples and positive class prevalence $\rho = 0.5$.

out of 30 base classifiers thirty times and calculated the mean and standard error of the mean AUC associated with the WOC ensemble. Figure 1b shows that the SUMMA ensemble increases in performance more readily as classifiers are added in the ensemble and saturates at a higher AUC than the WOC ensemble (Figure 1b). Moreover, we see that the SUMMA performance saturates at $n = 15$ classifiers suggesting that we can achieve a performance close to that of the full ensemble with only 15 classifiers (or 50% of all the base classifiers). We leave the investigation of finding an optimal number of classifiers for the ensemble construction for future research.




(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the number of base classifiers

(b) AUC of the SUMMA ensemble as a function of the number of base classifiers

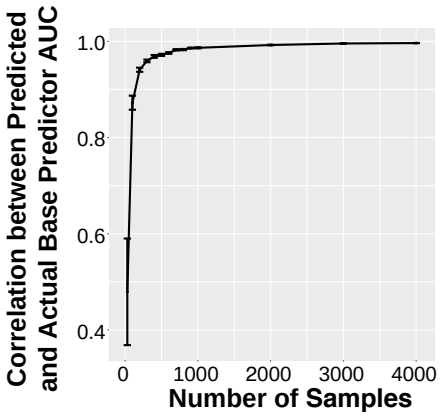
Figure 2: The dependence of the SUMMA results with the number of base classifiers for $N = 1000$ samples and $\rho = 0.5$.

In the second part of the analysis, we empirically tested the dependence of SUMMA on varying number of classifiers, samples, and class prevalence. In each case we change only one of the simulation parameter from their default values of, $M = 30$ base classifiers, $N = 1000$ samples, and the prevalence of class 1, $\rho = 1/2$. The error bars in the figures represents the standard error of the mean for 30 repeated experiments.

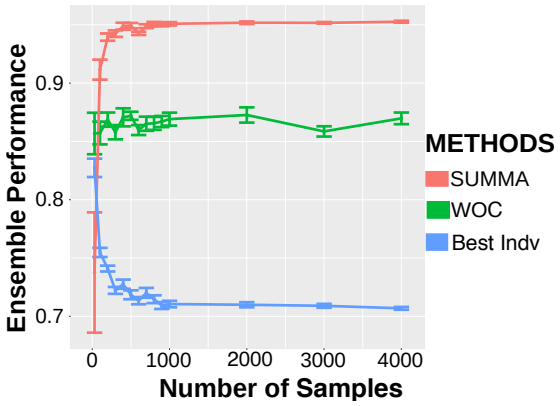
First, we tested the influence of the number of classifiers by simulating predictions and applying SUMMA to ensembles composed of $M = \{5, 6, 7, \ldots, 60\}$ base classifiers. This experiment is different from the analysis of Fig 1.b, because here we run SUMMA using $M = \{5, 6, 7, \ldots, 60\}$ (Fig. 2a) base classifiers as opposed to using the same covariance matrix of all 30 base classifiers as done in the experiments leading to Fig 1.b. In other words, the inference of weights is done for each set of $M$ classifiers as opposed to the earlier analysis. Intuitively, inferring $\Delta$ for larger covariance matrices should become more accurate because the number of equations grows faster, $M(M-1)/2$, than the number of parameters $\Delta_i$, $M$. Indeed, this intuition is confirmed in Figure 2. We see that the correlation between the predicted versus actual AUC of base classifiers inferred by SUMMA for $M = 5$ is $\approx$0.87, and increases readily to $\approx$0.97 for $M \geq 15$. We then tested how the number of classifiers affected the performance of the corresponding SUMMA ensemble. In Figure 2b we find that the SUMMA ensemble outperforms the WOC and best individual performing classifier in the ensemble. Moreover, as the number of classifiers increases, SUMMA's performance increases towards the perfect $AUC = 1$ even if the best individual classifier AUC never exceeds 0.75.

Next, we tested how the number of samples affect the performance of SUMMA. From the law of large numbers, one would expect that the performance of SUMMA would improve with the number of samples. This is simply because the error in estimating the covariance matrix elements decreases with $N$ which in turn results in a more reliably estimate of the AUCs of base classifiers. Indeed, in Figure 3a we see that the correlation between the SUMMA inferred AUC and the actual AUC of base classifiers monotonically increases from $\approx$0.57 to $\approx$1 when increasing $N$ from 30 to 4000 samples. Furthermore, the AUC of the SUMMA ensemble is also increasing with the number of samples mainly due to the fact that we can estimate the performance of base classifiers more reliably with increasing sample size (Figure 3a). Note that the WOC performance is not much affected by the sample size. When we have a very small number of samples ($N <= 50$) the uncertainty in the estimated performances of base classifiers is large enough to negatively influence the SUMMA ensemble performance as shown in Figure 3b. In such cases the WOC ensemble can be preferred to SUMMA. However, when ($N > 50$) the SUMMA ensemble performs significantly better than the WOC ensemble.

Lastly, we tested the influence of class prevalence on the performance of SUMMA. The accuracy of the SUMMA inferred AUC decreases for $\rho \leq$0.2 or $\rho \geq$0.8 (see Figure 4a). This is an intuitive result if we think that as the class prevalence moves to the extremes, the minority class has less samples, which results in larger errors in the estimation of minority class averages. The result is that the performance of the SUMMA ensemble in data with highly imbalanced classes is worse than in the case of balanced classes, albeit not by much as shown in Figure 4b. In this context it is relevant to note that high class imbalance also has a negative effect in the performance of supervised ensemble methods such as stacking (Padmaja et al., 2007).

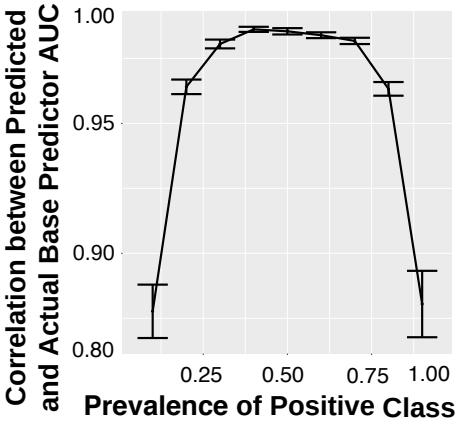



(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the number of samples

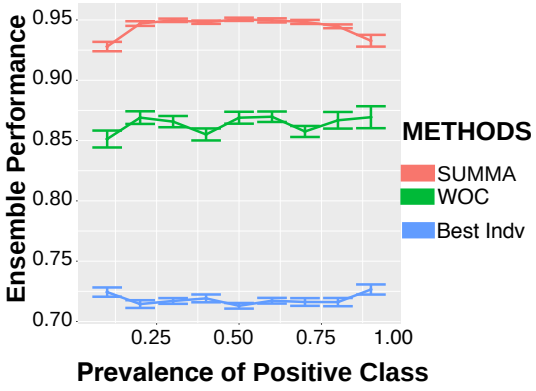



(b) AUC of the SUMMA ensemble as a function of the number of samples

Figure 3: The dependence of the SUMMA results with the number of samples for $M = 30$ base classifiers and positive class prevalence $\rho = 0.5$.




(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the prevalence of the positive class




(b) AUC of the SUMMA ensemble as a function of the prevalence of the positive class

Figure 4: The dependence of the SUMMA results with the prevalence of the positive class $\rho$ for $M = 30$ base classifiers and $N = 1000$ samples

### 7.2. Inference of genes targeted by a transcriptional regulator: The BCL6 DREAM Challange

Crowd-sourcing data competitions, such as Kaggle (www.kaggle.com) and the DREAM Challenges (www.dreamchallenges.org) (Prill et al., 2010; Saez-Rodriguez et al., 2016), have become effective at benchmarking a diverse array of machine learning methods while find-

ing efficient solutions to challenging real life problems. The diversity of machine learning strategies applied by challenge participants in these competitions results in a large number of independent predictions for the same test set which can be aggregated into an ensemble prediction with methods such as SUMMA. In this section we will use one such competition, the DREAM BCL6 challenge (Stolovitzky et al., 2009) to test the performance of SUMMA. DREAM Challenges lend themselves for this task because after the finalization of the competitions, DREAM organizers share the gold standard labels associated with the test set as well as the collection of participants' algorithms and predictions. This allowed us to objectively evaluate the performance of SUMMA in a problem of biological interest.

In the BCL6 challenge, participants were asked to infer whether the activity of a given gene is regulated, or in biological jargon: is targeted, by the transcription factor BCL6. The participants were provided an unlabeled feature matrix consisting of micro array measurements, with each element representing the relative abundance of RNA transcripts corresponding to a specific gene and given a specific perturbation. Additionally, they were allowed to incorporate any additional data that could be of use. They were then asked to report the confidence scores, in rank order for 200 genes, indicating whether a gene was a target of BCL6. Hidden from the participants were the experimentally determined class labels for each gene, in which 53 of the 200 ($\rho = 0.265$) genes were determined to be targets of BCL6. This way of creating gold standard labels is typical in biology and is very expensive both in monetary value as well as human resources. Therefore, the BCL6 challenge is a perfect application of SUMMA where there is no labeled data readily available for training.

Eleven teams participated and submitted predictions to this challenge. Therefore, the input to SUMMA was a matrix of size 200 genes by 11 methods, the elements of which are confidence scores between 0 and 1. We used the gold standard labels created by the organizers to compare the performance of SUMMA to that of individual methods, as well as to other methods including the Spectral Meta Learner (SML) algorithm (Parisi et al., 2014), the WOC and an unsupervised classifier based on k-means clustering.

We first tested the ability of SUMMA to estimate the AUCs of individual classifiers. As seen in Figure 5, SUMMA could reliably estimate the performance of individual classifiers with a correlation of 0.96 between the inferred and actual AUCs. Figure 5 also shows the AUC of the SUMMA ensemble, obtained by ranking samples according to the SUMMA score $c_k^{SUMMA}$ defined in (21). The SUMMA AUC (0.93) was significantly better than the WOC AUC (0.82) and the best individual method AUC (0.85). The success of the SUMMA ensemble in predicting the targets of BCL6 can be attributed in part to the fact that, due to the lack of labeled training data, challenge participants created a very diverse set of predictions based on biological interpretations of the data rather than, as is customary in other contexts, applying similar supervised models to the same training data for classifier generation. As a result the conditional independence assumption is not significantly violated, which allowed us to create relatively accurate estimates of the performances of base classifiers. Next we checked the effect of using the ranks rather than the confidence scores provided by the participants in the ensemble performance. For that we have used the AUCs of base classifiers inferred by SUMMA but formed the aggregate scores with the confidence scores generated by individual classifiers as opposed to converting them to ranks. This procedure resulted in an AUC of 0.83 as opposed to 0.93 obtained using SUMMA, which suggests a benefit in using rank transformation. The reason for this could be that, even

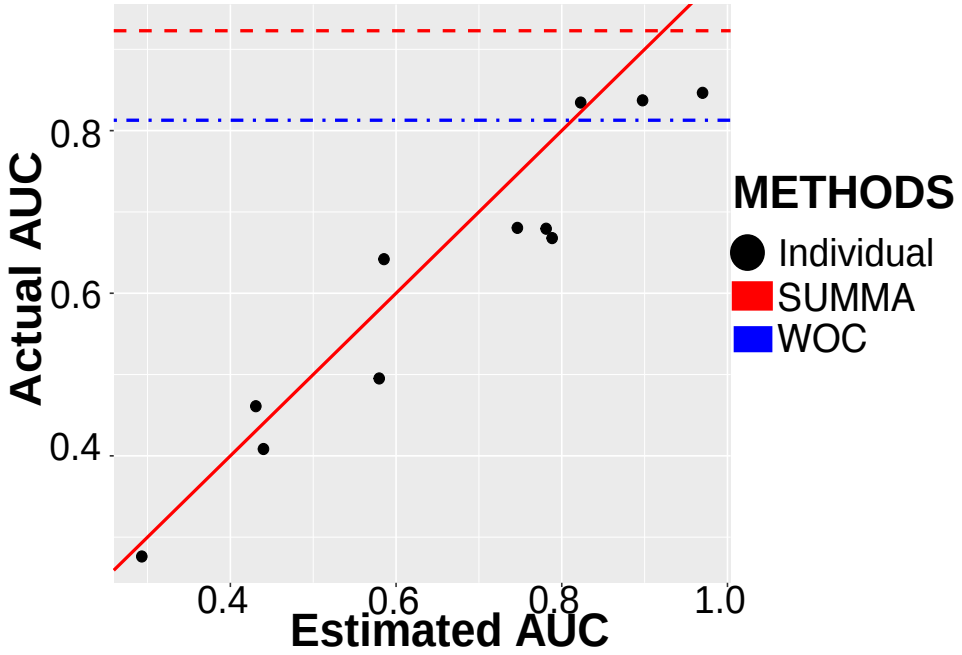



Figure 5: Comparison between the SUMMA inferred AUC on the BCL6 data and the actual AUC of the base 11 base classifiers

though confidence values are provided in the interval $[0, 1]$, it is likely that their distribution and interpretation is different in each prediction, which may have a negative impact in confidence score aggregation. Our results show that in situations in which we don't have control on the generation of confidence scores by classifiers or do not have labeled data to re-calibrate outputs of base classifiers, rank transformation allows for a robust way of normalizing confidence scores.

Next we investigate the classification performance of SUMMA. The results of this endeavor are summarized in Table 1. We calculated the balanced accuracy of SUMMA using the binary labels from the MLE estimate in (19) which resulted in a balanced accuracy (BA) of 0.82. For comparison we ran the SML algorithm in the same dataset by binarizing outputs of base classifiers. Since the challenge participants were not asked to provide a threshold, we binarized outputs of individual classifiers by using the natural threshold of 0.5 for the confidence. The SML had a BA of 0.80 whereas the best individual classifier had a BA of 0.79. We would also note that the second best method had a BA of 0.66 which shows that putting the right threshold is a challenging task that might lead to overfitting. As a further comparison, we also run $k$-means algorithm with $k = 2$ on the same dataset. In particular, we provided the matrix of confidence scores produced by base classifiers as input to the $k$-means algorithm. The $k-$means algorithm clustered the samples in to two clusters which we used to assign each sample a class label. The resulting classifier had a BA of 0.52 significantly lower than the other methods mentioned above.

| Method | Balanced Accuracy |
|---|---|
| SUMMA | 0.82 |
| SML | 0.8 |
| Best Individual | 0.79 |
| Second Best Individual | 0.66 |
| k-means (k=2) | 0.52 |

Table 1: The performance (balance accuracy) of ensemble methods and the two best individual methods used to solve the BCL6 DREAM Challenge

| Name | # Features | # Samples | Reference | Prevalence of Minority Class |
|---|---|---|---|---|
| Bank Marketing | 17 | 45211 | (Moro et al., 2014) | 0.11 |
| Ionosphere | 35 | 351 | (Sigillito et al., 1989) | 0.35 |
| Mammographic Mass | 6 | 830 | (Elter et al., 2007) | 0.48 |
| Parkinsons | 23 | 195 | (Little et al., 2007) | 0.24 |
| Yeast | 9 | 892 | (Nakai and Kanehisa, 1991) | 0.48 |

Table 2: Summary of Real World Data Sets from UCI Machine Learning Repository

### 7.3. Applying SUMMA to Real World Data in Diverse Domains

The main purpose of the analysis in this example is to study the robustness of SUMMA in typical real-life cases where our assumption of conditional independence is likely violated. For this purpose we trained a variety of base classifiers on the same training set and ran SUMMA on an independent test set. We applied the same procedure on five data sets coming from different fields (Table 2) taken from the UCI Machine Learning Repository (Lichman, 2013). More information about these five datasets can be found in Appendix E. With the exception of the Bank Marketing data, we divided each dataset into half, and used the first half to train the base classifiers and the second half to evaluate SUMMA. For the Bank Marketing data, which had 45,211 samples, we randomly selected 1000 samples for training and another 1000 samples for the evaluation. We restricted the sample size mainly to reduce computational burden of training base classifiers. We trained base classifiers using the R package caret (Kuhn et al., 2008b), which we chose for its ease of use, its inclusion of large diversity of popular classifiers, and its automatic layout for doing cross-validation (Kuhn et al., 2008a). However, we would like to emphasize that any base classifier such as the AUC maximizing algorithms OPAUC (Gao et al., 2013) can be used to form the SUMMA ensemble. We chose $M = 22$ base classifiers as shown in (Table 3), and used ten-fold cross validation for their training. We first tested whether SUMMA can reliably infer the AUC of each base classifier in the test set. In all the data sets considered, the correlation between the true AUC and the SUMMA inferred AUC of classifiers was above 0.75, as shown in Figure 6. This remarkable correlation in AUC values in these data sets is, however, lower than that obtained in the synthetic data (0.98) and the BCL6 challenge data

| Name | Main Method | RLibrary |
|---|---|---|
| adaboost | Adaboost | fastAdaboost |
| avNNet | Model Averaged Neural Network | nnet |
| bayesglm | Bayesian Generalized Linear Model | arm |
| ctree | Conditional Inference Tree | party |
| earth | Multivariate Adaptive Regression Spline | earth |
| gbm | Stochastic Gradient Boosting | gbm |
| glm | Generalized Linear Model | stats |
| glmnet | Lasso and Elastic-Net Regularized Generalized Linear Models | glmnet |
| J48 | C4.5-like Trees | RWeka |
| Jrip | Rule-Based Classifier | RWeka |
| C5.0 | Decision Trees and Rule-Based Models | C50 |
| knn | k-Nearest Neighbors | kknn |
| LMT | Logistic Model Trees | RWeka |
| mlp | Multi-Layer Perceptron | RSNNS |
| nb | Naive Bayes | klaR |
| nnet | Neural network | nnet |
| rf | Random Forest | randomForest |
| rpart | Recursive Partitioning and Regression Trees | rpart |
| simpls | Partial Least Squares | pls |
| svmLinear2 | Support Vector Machine with Linear Kernel | e1071 |
| svmRadial | Support Vector Machine with Radial Kernel | kernlab |
| xgbLinear | eXtreme Gradient Boosting | xgboost |
| xgbTree | eXtreme Gradient Boosting | xgboost |

Table 3: Machine learning methods used in this Section

(0.96). This is likely due to the conditional dependence between base classifiers. Table 4 shows the ranking of classifiers including SUMMA ensemble in terms of AUC. We can easily observe that classifiers that perform best in one data set do not necessarily perform well in other data sets. In fact, they can be one of the worst in other data sets. This is most likely due to the distributions of the data being different in different data sets and classifiers with different theoretical backgrounds are more suitable to be applied in one type of data than other. In comparison, SUMMA performs better than the best individual classifier in the Bank Marketing, Parkinsons and Yeast datasets, and second best in the mammographic masses and fifth in ionosphere data sets. Next, for two of the datasets, Bank Marketing and Yeast, we analyzed how the number of integrated classifiers affects the performance of SUMMA prediction by examining randomly sampled combinations of individual classifiers (Figures 7a and b respectively). SUMMA performs better than individual classifiers even when integrating small sets of individual predictions. Performance increases further with the number of integrated classifiers. For instance, for 15 randomly selected individual classifiers, the SUMMA ensemble performs better than the best amongst the 15 classifiers in 98% of the cases in the Bank Marketing data set and it ranks best in 80% of the cases and best or second best in 97% of the cases in the Yeast data set demonstrating the robustness of SUMMA. Table 5 shows the frequency with which SUMMA outperforms the WOC ensemble prediction in the Bank Marketing and Yeast data. For example, for the Bank Marketing data set if we combine random 10 teams, 99% of the times SUMMA performs better than WOC. For the Yeast data, SUMMA gives a better prediction than WOC in about 65% of the times.

We next compared the classification accuracy of SUMMA to individual classifiers. For that we used the binary label predictions output by caret and computed the balanced accuracy of each individual classifier. Table 4 shows the ranking of classifiers including SUMMA ensemble in terms of their balanced accuracies. Similar to the rank based outputs, SUMMA is better than the best in 3 of the 5 datasets and is the second best in the rest. Moreover, as before classifiers that perform best in one data set do not necessarily perform well in other data sets.

These results highlight the robustness of SUMMA even in real applications where the conditional independence assumption is not guaranteed. As noticed above, the best team in any one example could perform relatively poorly or even amongst the worst in other examples. However, SUMMA is amongst the top in all examples, suggesting that using SUMMA is a sound strategy for aggregation when many predictions to the same problem are available and one do not have any strong prior belief on the top performing algorithm.

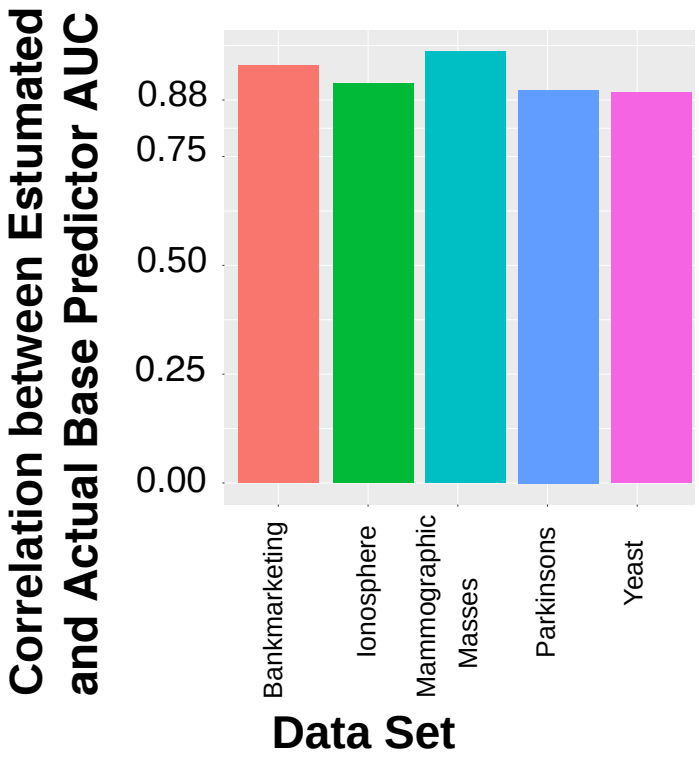
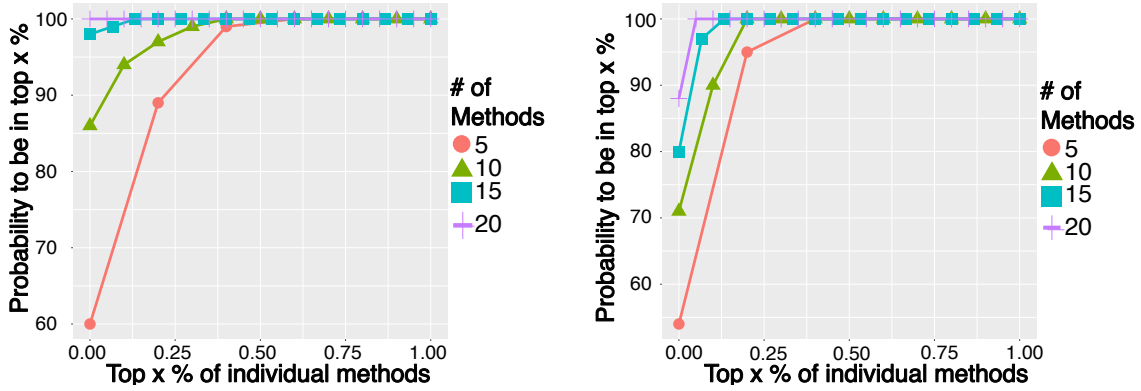



Figure 6: Correlation between the actual and estimated AUC of the base classifiers tested on the six UCI datasets considered in this paper.




(a) Bank Marketing Data

(b) Yeast Data

Figure 7: Robustness of SUMMA with increasing number base classifiers (Y axis represents how often SUMMA is in the top $x$% of methods)

## 8. Conclusions

In this paper, we introduced a novel methodology for unsupervised ensemble learning: the Strategy for Unsupervised Multiple Method Aggregation (SUMMA). Under the assumption

| Method | Bank Marketing | Ionosphere | Mammographic Mass | Parkinsons | Yeast |
|---|---|---|---|---|---|
| Earth | 2 | 22 | 5 | 6 | 6 |
| svmRadial | 12 | 1 | 18 | 5 | 5 |
| gbm | 5 | 7 | 1 | 4 | 4 |
| C5.0 | 8 | 8 | 7 | 2 | 21 |
| rf | 7 | 3 | 6 | 5 | 2 |
| SUMMA | 1 | 5 | 2 | 1 | 1 |

Table 4: Ranking of the different methods in each application domain. Only the classifiers that ranked first in at least one application are listed.

| Number Classifiers | % SUMMA is better than WOC | |
|---|---|---|
| | Bank Marketing Data | Yeast Data |
| 5 | 90 | 69 |
| 10 | 99 | 68 |
| 15 | 100 | 63 |
| 20 | 100 | 68 |

Table 5: Percentage of times SUMMA outperforms WOC.

| Method | Bank Marketing | Ionosphere | Mammographic Mass | Parkinsons | Yeast |
|---|---|---|---|---|---|
| glm | 2 | 14 | 12 | 9 | 12 |
| svmRadial | 15 | 1 | 11 | 7 | 10 |
| simpls | 12 | 10 | 1 | 2 | 11 |
| rf | 10 | 3 | 4 | 6 | 2 |
| SUMMA | 1 | 2 | 2 | 1 | 1 |

Table 6: Ranking of Methods using classification performance (Balanced Accuracy) in each application domain. Only the classifiers that ranked first in at least one application are listed.

that base classifiers assign ranks independently of each other to samples within each class (conditional independence assumption) we showed that the SUMMA algorithm can infer the AUC of base classifiers from the covariance of their rank predictions in the absence of labeled data. We then used the inferred AUCs as constraints in the context of a maximum entropy inference of the probability of the class of a sample given its rank. This probability was then used in a maximum likelihood estimation to derive a score that integrated the ranking assigned by each base classifier to each sample, which allowed us to predict the

class labels most likely associated with each sample, as well as a new integrated ranking for the samples from which we can compute the ensemble AUC.

We evaluated the performance of SUMMA on synthetic data, on the predictions submitted to a crowd-sourced competition (the BCL6 DREAM Challenge), and on multiple predictions to classification problems arising in real life contexts. The application of SUMMA in different settings shows that SUMMA can reliably estimate the AUCs of base classifiers in idealized problems where the assumption of conditional independence holds, as well as in real life applications where the assumption of conditional independence is likely violated. SUMMA performed better than the best base classifiers in the synthetic data as well as the BCL6 DREAM Challenge. In a third application, we used five datasets available from UCI machine learning repository which we split into a training set to train base classifiers and a test set to run ensemble methods including SUMMA. Our results show that SUMMA performs better than the best performer in three of the five datasets and among the top performers in the other cases.

In this paper we have introduced SUMMA for a binary classification problem. SUMMA can be extended to multi-class classification problems as follows. Assume that each classifier in an ensemble provides a score (such as confidence level or the distance to a surface) associated to the assignment of each sample to each of $k > 2$ classes. For each class $i$ we run SUMMA as a binary classification problem, with class $i$ being the positive class and all the other $k-1$ classes constituting the negative class. In this way, for each class $i$ and each sample $j$ we have a SUMMA score $S_{ij}$. A multi-class classification extension of SUMMA would consist of assigning sample $j$ the class $i$ that makes $S_{ij}$ maximum. This approach requires that the classifiers assign class scores independently given the positive and negative class in each of the $k$ binary problems.

SUMMA could be extended to cases where the conditional independence assumption between the classifiers does not hold and there is some structured correlation in the predictions. An example could be a block structure where a group of classifiers are correlated with each other but uncorrelated with predictors in other groups. A first step in that direction would be developing a method to identify the block structure and then average the predictions in each block before running SUMMA.

Another extension of our work would be to develop a modified version of SUMMA for the so-called rank-aggregation problem which is a very active area of research (Bhowmik and Ghosh, 2017). In the rank-aggregation problem, there is an unknown but true ordering of samples and each method is trying to predict this unknown ordering. The performance of a method is measured using various quantities such as *Kendall-tau* distance (Klementiev et al., 2008). In fact, one of the well established baseline methods used is Borda-counting which is strongly connected to the SUMMA score as we showed in the manuscript (Bhowmik and Ghosh, 2017). The first step towards this direction is determining the right performance metric to be used and setting up the right assumptions under which it can be estimated.

## References

Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

Javed A Aslam and Mark Montague. Models for metasearch. In *Proceedings of the ACM SIGIR Conference*, 2001.

Akshay Balsubramani and Yoav Freund. Scalable semi-supervised aggregation of classifiers. In *Advances in Neural Information Processing Systems*, 2015.

Rina Foygel Barber and Wooseok Ha. Gradient descent with non-convex constraints: local concavity determines convergence. *Information and Inference*, 7(4):755–806, 2018.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. On the effect of calibration in classifier combination. *Applied Intelligence*, 38(4):566–585, 2013.

Avradeep Bhowmik and Joydeep Ghosh. Letor methods for unsupervised rank aggregation. In *Proceedings of the International Conference on World Wide Web*, 2017.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the International Conference on Machine learning*, 2004.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C*, 28 (1):20–28, 1979.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39 (1):1–22, 1977.

Thomas G Dietterich. Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, 2:110–125, 2002.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

M Elter, R Schulz-Wendtland, and T Wittenberg. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, 34(11):4164–4172, 2007.

Peter Emerson. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358, 2013.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International Conference on Machine Learning*, 2013.

Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *Proceedings of International Conference on Machine Learning and Applications*, 2008.

Frank E Harrell. Ordinal logistic regression. In *Regression Modeling Strategies*, pages 331–343. Springer, 2001.

Ariel Jaffe, Boaz Nadler, and Yuval Kluger. Estimating the accuracies of multiple classifiers without labeled data. In *Artificial Intelligence and Statistics*, 2015.

Prateek Jain and Sewoong Oh. Learning mixtures of discrete product distributions using spectral decompositions. In *Conference on Learning Theory*, 2014.

Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4): 620, 1957.

Azam Karami, Mehran Yazdi, and Grégoire Mercier. Compression of hyperspectral images using discerete wavelet transform and tucker decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):444–450, 2012.

Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggregation with distance-based models. In *Proceedings of the International Conference on Machine Learning*, 2008.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Max Kuhn et al. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008a.

Max Kuhn et al. Caret package. *Journal of Statistical Software*, 28(5):1–26, 2008b.

M. Lichman. Uci machine learning repository, 2013.

Max A Little, Patrick E McSharry, Stephen J Roberts, Declan AE Costello, and Irene M Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering Online*, 6(1):23, 2007.

Tie-Yan Liu, Hang Li, and Yu-Ting Liu. Supervised rank aggregation based on rankings, 2010. US Patent 7,840,522.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Andrej Aderhold, Richard Bonneau, Yukun Chen, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796, 2012.

Caren Marzban. The roc curve and the area under it as performance measures. *Weather and Forecasting*, 19(6):1106–1114, 2004.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Kenta Nakai and Minoru Kanehisa. Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins: Structure, Function, and Bioinformatics*, 11(2): 95–110, 1991.

Alexandru Niculescu-Mizil, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang, Jing Xiao, Jianying Hu, Moninder Singh, et al. Winning the kdd cup orange challenge with ensemble selection. In *Proceedings of the International Conference on KDD-Cup*, 2009.

Shmuel Nitzan and Jacob Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, pages 289–297, 1982.

Shmuel Nitzan and Ariel Rubinstein. A further characterization of borda ranking method. *Public Choice*, 36(1):153–158, 1981.

T Maruthi Padmaja, Narendra Dhulipalla, Raju S Bapi, and P Radha Krishna. Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection. In *Proceedings of the International Conference on Advanced Computing and Communications*, 2007.

Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PloS One*, 5(2):e9202, 2010.

Julio Saez-Rodriguez, James C Costello, Stephen H Friend, Michael R Kellen, Lara Mangravite, Pablo Meyer, Thea Norman, and Gustavo Stolovitzky. Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nature Reviews Genetics*, 17(8): 470, 2016.

Robert E Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Amartya Sen. Social choice theory. *Handbook of Mathematical Economics*, 3:1073–1181, 1986.

Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

Gustavo Stolovitzky, Robert J Prill, and Andrea Califano. Lessons from the dream2 challenges: a community effort to assess biological network inference. *Annals of the New York Academy of Sciences*, 1158(1):159–195, 2009.

Karthik Subbian and Prem Melville. Supervised rank aggregation for predicting influencers in twitter. In *Proceedings of the International Conference on Privacy, Security, Risk and Trust*, 2011.

Liang Sun, Tomonori Honda, Vesselin Diev, Gregory Gancarz, Jeong-Yoon Lee, Ying Liu, Mona Mahmoudi, Raghav Mathur, Shahinur Rahman, Steve Wickert, et al. Maximize auc in default prediction: Modeling and blending.

James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

Chih-Fong Tsai and Yu-Chieh Hsiao. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1):258–269, 2010.

Lieven Vandenberghe and Stephen Boyd. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29(3):283–299, 1999.

Sean Whalen and Gaurav Pandey. A comparative analysis of ensemble classifiers: case studies in genomics. In *Proceedings of the International Conference on Data Mining*, 2013.

David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

## Appendix A. Details of SDP Approach

### A.1. Semi-Definite Programming

We start this section by the following definition.

**Definition 9** *We say that a matrix $Q$ is off-diagonal rank-one if there exists a vector $q$ and a diagonal matrix $D$ s.t.*

$$Q = qq^T + D. \tag{28}$$

In order to show that $R$ is the unique solution to the problem (11), we prove a lemma stating that a unique solution of the problem (11) exists for an arbitrary matrix $Q$ whose off-diagonal elements coincide with a rank one matrix. The application of this lemma with $Q = \Sigma_2$ will then allow us to show the desired result.

**Lemma 10** *Let $Q$ be an off-diagonal rank-one square matrix of size $M$ such that*

$$Q = qq^T + D_0, \tag{29}$$

*for some $q \in \mathbb{R}^M$ with $M \geq 3$, and diagonal matrix $D_0$ and assume that $q_1, q_2, q_3 \neq 0$. Then the optimization problem*

$$\min_D rank(Q - D) \quad s.t. \quad D(i,j) = 0 \quad for \quad i \neq j, \quad and \quad Q - D \succeq 0, \tag{30}$$

*has the unique solution $D_0$, with $Q - D_0 = qq^T = (-q)(-q)^T$ so that we can recover $q$ up to its sign.*

**Proof** Let $Q$ be defined as in Equation (29) with $q_1, q_2, q_3 \neq 0$ and $D_0$. Since $Q(i,j) = q_i q_j \neq 0$, it is obvious that for any diagonal matrix $D$, $Q - D \neq 0$. Hence, $rank(Q - D) > 0$ for any diagonal matrix $D$. Moreover, since $Q - D_0 = qq^T$ one solution of the optimization problem in Equation (11) is $D_0$. Next, we show that $D_0$ is the only diagonal matrix that is a feasible point of the optimization problem.

Suppose there exists a diagonal matrix $D_s$ such that $D_s \neq D_0$ and $rank(Q - D_s) = 1$. Then since $Q - D_s$ is symmetric, there exists $s \in \mathbb{R}^M$ such that $Q - D_s = ss^T$ and $q \neq s$. Since both $D_0$ and $D_s$ are diagonal and $D_s + ss^T = Q = D_0 + qq^T$, then the equality

$$q_i q_j = s_i s_j. \tag{31}$$

must be true for all $i \neq j$ where $i, j \in \{1, 2, \ldots, M\}$.

Without loss of generality lets assume that $q_1 \neq 0 > s_1 \neq 0$. Equation (31) for $(i,j) = (1,2)$ implies that $s_2 > q_2$, and for $(i,j) = (1,3)$ implies that $s_3 > q_3$. Under these inequalities, when $(i,j) = (2,3)$ we see that $s_2 s_3 > q_2 q_3$, which contradicts equation (31). Therefore, $q_1 = s_1$ which from Equation (31) implies that for all $i, q_i = s_i$. Therefore, $D_0 = D_s$ and consequently $D_0$ is the unique solution to the optimization problem given in equation (11). ■

As we discussed in the main text the optimization problem in (30) can be relaxed to the following SDP:

$$\min_D \quad \|Q - D\|_* \quad \text{s.t.} \quad D \text{ is diagonal, and} \quad Q - D \succeq 0. \tag{32}$$

**Theorem 11** *Suppose that $Q$ is an $M \times M$ matrix of the form given in Equation (29) for some $q$ and diagonal matrix $D_0$, then the optimization problem in Equation (32) has the unique solution $D_0$, provided that*

$$q_i^2 < \sum_{j \neq i} q_j^2, \quad \forall i \in \{1, 2, \ldots, M\}. \tag{33}$$

*Moreover, $Q - D_0 = qq^T$ so that we can recover $q$ up to its sign.*

**Proof** Let $D$ be an arbitrary diagonal matrix such that $Q - D$ is a PSD (Positive Semidefinite) matrix. Then the eigenvalues of $Q - D$ are non-negative and equal to the singular values of $Q - D$. Combined with the fact that the trace of a matrix is equal to the sum of its eigenvalues, we know the following:

$$\|Q - D\|_* = Tr(Q - D).$$

Without loss of generality lets assume that the diagonal entries of $Q$ are equal to 0, that is $Q = qq^T - diag(qq^T)$. In this case, note that $D_0 = -diag(qq^T)$ is a feasible solution for the optimization problem in Equation (13). Next let $D$ be an arbitrary solution and since $Q - D$ is PSD, $D_{ii} \leq 0$ for all $i$. Suppose for some $i$ we have $D_{ii} > -q_i^2$ and for all $j \neq i$ we have $D_{jj} \geq -q_j^2$. For $j \neq i$, consider the following sub-matrix of $Q - D$,

$$(Q - D)^{ij} = \begin{bmatrix} -D_{ii} & q_i q_j \\ q_i q_j & -D_{jj} \end{bmatrix}.$$

Since $det((Q - D)^{ij}) = D_{ii}D_{jj} - q_i^2 q_j^2 < 0$, the submatrix $(Q - D)^{ij}$ of $Q - D$ is negative definite which contradicts the fact that $Q - D$ is PSD. Therefore, combined with the fact that $D_{ii} \leq 0$ for each $i$, this implies that in order for a diagonal matrix $D$ to be a feasible point for the optimization problem (13)

1. Either for all $i$, we have $D_{ii} \leq -q_i^2$
2. Or there exist $i$ such that $D_{ii} > -q_i^2$, and $\forall j \neq i$ $D_{jj} \leq -q_j^2$.

Suppose $D$ is a feasible point that satisfies condition 1, then $Tr(Q - D) \geq Tr(Q + diag(qq^t))$ for every feasible $D$. Hence, $D_0 = -diag(qq^t)$ remains to be the unique solution of the optimization problem. Now suppose there exists a feasible point $D$ such that condition 2 of above is satisfied. Then WLO assume that $D_{11} > -q_1^2$ and $D_{jj} \leq -q_i^2$ for $j \geq 2$. Next, for each $j \geq 2$ consider the following submatrix:

$$(Q - D)^{1j} = \begin{bmatrix} -D_{11} & q_1 q_j \\ q_1 q_j & -D_{jj} \end{bmatrix}.$$

In order for $Q - D$ to be PSD, we should have $det((Q - D)^{1j}) = D_{11}D_{jj} - q_1^2 q_j^2 > 0$, which in turn implies that

$$D_{jj} < \frac{q_1^2 q_j^2}{D_{11}}. \tag{34}$$

Using Equation (34), we observe that

$$\begin{aligned} \|Q-D\|_* &= Tr(Q-D) = -D_{11} - \sum_{j>1} D_{jj} \\ &> -D_{11} + \sum_{j>1} -\frac{q_1^2 q_j^2}{D_{11}}, \end{aligned} \tag{35}$$

where Equation (35) comes from Equation (34). Now from the assumption that $q_1^2 < \sum_{j>1} q_j^2$, we have $D_{11} > -q_1^2 > -q_1\sqrt{\sum_{j>1} q_j^2}$, and for any $D_{11} \in [-q_1^2, 0]$

$$\|Q-D\|_* < -D_{11} - \sum_{j>1} \frac{q_1^2 q_j^2}{D_{11}} > -q_1^2 - \sum_{j>1} q_j^2 = \left\|Q + diag(qq^t)\right\|_*. \tag{36}$$

A generalization of the above argument implies that if for each $i$, $q_i^2 < \sqrt{\sum_{j\neq i} q_j^2}$, then $\|Q-D\|_* > \left\|Q + diag(qq^t)\right\|_*$ so that $D_0 = -diag(qq^t)$ the unique solution to the optimization problem (13). ■

Our next result shows that the solutions of (30) and (32) coincide.

## Appendix B. Proof of Theorem 2

**Proof** For simplicity, we denote the set of $M$ classifiers $\{g_1, \cdots, g_M\}$ as $\{1, ..., M\}$ and proceed by induction on $n$. Note that due to our problem setup for each $i$, we have $\mathbb{E}[R_i] = N/2$. Suppose the classifiers $\{g_i\}_{i=1}^M$ are $l^{th}$ order conditionally independent. The conditional independence assumption implies that any given subset of classifiers of size $n \leq l$ of $\{1, ..., M\}$ are $n^{th}$ order conditionally independent. Next, let $n = 2$, and suppose that the two methods 1 and 2 are conditionally independent as such equation (1) is satisfied with $n = 2$. Then by using law of total expectation, we obtain the following:

$$\begin{aligned} \Sigma_2(1,2) &= \mathbb{E}[(R_1 - \mathbb{E}[R_1])(R_2 - \mathbb{E}[R_2])] = \mathbb{E}[R_1R_2] - \mathbb{E}[R_1]\,\mathbb{E}[R_2] \\ &= \mathbb{E}[R_1R_2|y=1]\rho + \mathbb{E}[R_1R_2|y=0](1-\rho) - \mathbb{E}[R_1]\,\mathbb{E}[R_2] \\ &= \mathbb{E}[R_1|y=1]\,\mathbb{E}[R_2|y=1](\rho) + \mathbb{E}[R_1|y=0]\,\mathbb{E}[R_2|y=0](1-\rho) \\ &- \mathbb{E}[R_1]\,\mathbb{E}[R_2] \end{aligned} \tag{37}$$

where as defined before $\rho$ is the prevalence of class $y$=1. Similarly, from the law of total expectation we have

$$\begin{aligned} \mathbb{E}[R_1] &= \mathbb{E}[R_1|y=1]\rho + \mathbb{E}[R_1|y=0](1-\rho). \\ \mathbb{E}[R_2] &= \mathbb{E}[R_2|y=1]\rho + \mathbb{E}[R_2|y=0](1-\rho). \end{aligned} \tag{38}$$

Substituting Equation (38) into Equation (37), we obtain

$$\Sigma_2(1,2) = C(\rho)\Delta_1\Delta_2.$$

where $C(\rho) = \rho(1-\rho)$. To ease the notation for the inductive step, for any $n \geq 1$, let the random variable $S_n$ be defined as follows

$$S_n = (R_1 - \mathbb{E}[R_1]) \cdots (R_n - \mathbb{E}[R_n]) = (R_n - \mathbb{E}[R_n])S_{n-1},$$

where $S_0 = 1$. Also note that $\Sigma_n(1, \cdots, n) = \mathbb{E}[S_n]$ Next, we inductively show two formulas which are required for the proof of the theorem.
**Claim 1:** For any $1 \leq n \leq l \leq M$, $\mathbb{E}[S_n|y=0] = \rho^n \prod_{i=1}^n \Delta_i$.
**Proof:** For $n = 1$,

$$\begin{aligned}\mathbb{E}[S_1|y=0] &= \mathbb{E}[R_1 - \mathbb{E}[R_1]|y=0] = \mathbb{E}[R_1|y=0] - \mathbb{E}[R_1] \\ &= \mathbb{E}[R_1|y=0] - \rho\,\mathbb{E}[R_1|y=1] - (1-\rho)\,\mathbb{E}[R_1|y=0] \\ &= \rho(\mathbb{E}[R_1|y=0] - \mathbb{E}[R_1|y=1]) = \rho\Delta_1. \end{aligned} \tag{39}$$

Next assume the claim is true for $n-1$, then from the inductive hypothesis we have

$$\mathbb{E}[S_{n-1}|y=0] = \rho^{n-1} \prod_{j=1}^{n-1} \Delta_j. \tag{40}$$

Let us now prove that if it is true for $n-1$, then it is true for $n$. From conditional independence and total law of expectation we get

$$\begin{aligned}\mathbb{E}[S_n|y=0] &= \mathbb{E}[(R_n - \mathbb{E}[R_n])S_{n-1}|y=0] = (\mathbb{E}[R_n|y=0] - \mathbb{E}[R_n])\,\mathbb{E}[S_{n-1}|y=0] \\ &= \mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - \mathbb{E}[R_n]\,\mathbb{E}[S_{n-1}|y=0] \\ &= \mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - (1-\rho)\,\mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - \rho\,\mathbb{E}[R_n|y=1]\,\mathbb{E}[S_{n-1}|y=0] \\ &= \rho\,\mathbb{E}[S_{n-1}|y=0](\mathbb{E}[R_n|y=0] - \mathbb{E}[R_n|y=1]) \\ &= \rho\,\mathbb{E}[S_{n-1}|y=0]\Delta_n = \rho\Delta_n\rho^{n-1}\prod_{j=1}^{n-1}\Delta_j = \rho^n\prod_{j=1}^{n}\Delta_j, \end{aligned}$$

where the previous to last equality follows from inductive assumption and completes the proof.

**Claim 2:** For any $1 \leq n \leq l \leq M$, $\mathbb{E}[S_n|y=1] = (\rho-1)^n \prod_{i=1}^n \Delta_i$.
The proof of this claims follows from Claim 1 by replacing $\rho \to 1-\rho$ and $\Delta_i \to -\Delta_i$.
Now we are ready to prove the main theorem. Using law of total expectation and claim 1 and 2, we obtain the following set of equations

$$\begin{aligned}\mathbb{E}[S_n] &= \mathbb{E}[S_n|y=1]\rho + \mathbb{E}[S_n|y=0](1-\rho) = (\rho(\rho-1)^l + (1-\rho)\rho^n)\prod_{j=1}^{n}\Delta_j \\ &= \rho(1-\rho)(\rho^{n-1} - (\rho-1)^{n-1})\prod_{j=1}^{n}\Delta_j, \end{aligned} \tag{41}$$

which completes the proof of the theorem. ■

## Appendix C. Proof of Theorem 4

**Proof** Let $N_1$ denote the positive samples and $N_0 = N - N_1$ denote the negative samples. Given a ranking of samples, the receiver operating characteristic (ROC) curve consists of

points representing the False Positive Rates ($\text{FPR}_i$) and the True Positive Rates ($\text{TPR}_i$) empirically evaluated for each threshold $i \in \{1, ..., N\}$. The area under the ROC (AUC) can be calculated according to the rectangle rule using the following formula:

$$AUC_i = \sum_{i=0}^{N-1} \text{TPR}_i \left(\text{FPR}_{i+1} - \text{FPR}_i\right), \tag{42}$$

where $FPR_0 := 0$ and $TPR_0 := 0$. The elements of the sum behave as follows. If the $i^{th}$ ranked sample has a positive label then the $\text{FPR}_{i+1} = \text{FPR}_i$, and if it has a negative label then $FPR_{i+1} = FPR_i + 1/N_0$. Moreover, if we let $\{r_{i_1}, \cdots, r_{i_{N_0}}\}$ denote the ranks of negative samples, then for any threshold $i_l$ the number of true positives is given as $i_l - l$ and consequently $TPR_l = (i_l - l)/N_1$. Using these observations, Equation (42) becomes

$$\begin{aligned} AUC_i &= \sum_{i=1}^{N} \text{TPR}_i(\text{FPR}_{i+1} - \text{FPR}_i) \\ &= \sum_{l:y_l=0} \frac{\text{TPR}_l}{N_1}\frac{1}{N_0} = \sum_{l=1}^{N_0} \frac{(r_{i_l} - l)}{N_1}\frac{1}{N_0} = \frac{\mathbb{E}[\widehat{R}_i|y=0]}{N_1} - \frac{N_0+1}{2N_1}. \end{aligned} \tag{43}$$

Next we express $\widehat{\Delta}_i$ in terms of $\mathbb{E}[\widehat{R}_i|y=0]$:

$$\begin{aligned} \frac{\widehat{\Delta}_i}{N} &= \frac{\frac{\sum_{i:y_k=0} r_{ik}}{N_0} - \frac{\sum_{k:y_k=1} r_{ik}}{N_1}}{N} \\ &= \frac{\frac{\sum_{k:y_k=0} r_{ik}}{N_0} + \frac{\sum_{k:y_k=0} r_{ik}}{N_1} - \frac{\sum_{i=1}^{N} r_{ik}}{N_1}}{N} = \frac{\mathbb{E}[\widehat{R}_i|y=0]_i}{N_1} - \frac{N+1}{2N_1} \\ \implies \frac{\mathbb{E}[\widehat{R}_i|y=0]_i}{N_1} &= \frac{\widehat{\Delta}_i}{N} + \frac{N+1}{2N_1}. \end{aligned} \tag{44}$$

If we substitute Equation (44) into Equation (43), we obtain

$$\begin{aligned} AUC_i &= \frac{\mathbb{E}[\widehat{R}_i|y=0]}{N_1} - \frac{N_1+1}{2N_1} = \frac{\widehat{\Delta}_i}{N} + \frac{N+1}{2N_1} - \frac{N_0+1}{2N_1} \\ &= \frac{\widehat{\Delta}_i}{N} + \frac{1}{2}, \end{aligned} \tag{45}$$

■

which completes the proof of the theorem.

## Appendix D. Proof of Lemma 6

**Proof** Consider the $k^{th}$ sample assigned rank $r_{ik}$ by the $i^{th}$ base classifier. Let $P_i(y_k|r_{ik})$ be the probability distribution that is *a priori* unknown and that is an extremum of the maximum entropy functional

$$J = -\sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} P(r_{ik}) P_i(y_k|r_{ik}) \log(P_i(y_k|r_{ik})) + \sum_{j=0}^{2} \lambda_{ij} \Lambda_j,$$

subject to the constraints $\Lambda_j$ for $j = (0, 1, 2)$, with

$$\Lambda_0 = N - \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} P_i(y_k|r_{ik}), \tag{46a}$$

$$\Lambda_1 = N\rho - \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}), \tag{46b}$$

$$\Lambda_2 = N\rho\, \mathbb{E}[R_i|y_k = 1] - \sum_{k=1}^{N} r_{ik} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}), \qquad \text{or equivalently}$$

$$= N\rho\big(\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - \sum_{k=1}^{N} r_{ik} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}). \tag{46c}$$

Here we constrain the inferred function such that the sum over labels normalizes to 1 (Equation 46a), the average occurrence of class one samples is reflective of its occurrence in the population ($N_1$) (Equation 46b), and lastly that the average rank condition by the class label recovers our empirical estimate of $\Delta_i$ (Equation 46c).

We consider the maximum entropy criterion satisfied when the variation of the functional $\delta J$ with respect to $P_i(y_k|r_{ik})$ is stationary,

$$\delta J = \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} \left[ -\frac{1}{N} \log\left(P_i(y_k|r_{ik})\right) - \frac{1}{N} - \lambda_{i0} - \lambda_{i1} y_k - \lambda_{i2} y_k r_{ik} \right] \delta P_i(y_k|r_{ik}) = 0$$

and consequently,

$$P_i(y_k|r_{ik}) = \frac{1}{Z} e^{-\lambda_{i1} y_k - \lambda_{i2} y_k r_{ik}}. \tag{47}$$

In Equation (47), $\lambda_{i0}$ is incorporated in the constant $Z$, which satisfies the normalization constraint in Equation (46a). Therefore, $Z = 1 + e^{-\lambda_{i1} - \lambda_{i2} r_{ik}}$.

Next, we solve for the remaining Lagrange multipliers by substituting Equation (47) into Equations (46b, 46c). Each of these equations, after summing over class labels, amount to calculating sums over $P_i(y_k = 1|r_{ik}) = (1 + e^{\lambda_{i1} + \lambda_{i2} r_{ik}})^{-1}$. Here, the functional form of $P_i(y_k = 1|r_{ik})$ precludes clear analytical solutions to these summations and consequently precludes solutions for $\lambda_{i1}$ and $\lambda_{i2}$. To circumvent this challenge we approximate the distribution $P_i(y_k = 1|r_{ik})$ by its Taylor series in $\lambda_{i1}$ and $\lambda_{i2}$ about their respective values, $\lambda_1^*$ and $\lambda_2^*$, representative of an uninformative classifier.

Consider an uninformative classifier as one that assigns each sample a rank without regard to its true class label. That is, it assigns the $N_1$ positive samples and the $N_0$ negative samples to be classified, random ranks between 1 and $N$. As the probability of such classifier to locate a positive class samples at rank $r$ is independent of rank, then $\lambda_2^* = 0$ and given that there are $N_1 = \rho N$ positive class items in the sample, $\lambda_1^* = \log((1-\rho)/\rho)$, where the asterisk explicitly indicates that the parameters are those of the uninformative classifier. For clarity in exposition, we will explicitly include the Lagrange multipliers in the notation of the probability of the class of an item at a given rank as $P_i(y_k|r_{ik}, \lambda_{i1}, \lambda_{i2})$;

then by substituting these parameters into the maximum entropy distribution we see that the uninformative classifier,

$$P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) = \left(1 + e^{\lambda_1^* + \lambda_2^* r_{ik}}\right)^{-1} = \rho.$$

The Taylor series of $P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})$ about the uninformative classifier amounts to expanding $P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})$ in $\lambda_{i1}$ and $\lambda_{i2}$ about $\lambda_1^*$ and $\lambda_2^*$. We assume that all considered classifiers are weakly predictive, so that the terms of order two and higher negligibly contribute the approximated function. Then the Taylor series to first order is,

$$\begin{aligned} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2}) \approx P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) \\ + \partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} \delta\lambda_{i1} \\ + \partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} \lambda_{i2} \end{aligned} \quad (48)$$

with $\partial_{\lambda_{i1}}$ and $\partial_{\lambda_{i2}}$ being the partial derivatives $\frac{\partial}{\partial\lambda_{i1}}$ and $\frac{\partial}{\partial\lambda_{i2}}$, and $\delta\lambda_{i1} = \lambda_{i1} - \log((1-\rho)/\rho)$. The partial derivatives are as follows,

$$\begin{aligned} \partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} &= -P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\left(1 - P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\right), \\ \partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} &= -P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\left(1 - P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\right) r_{ik}, \end{aligned}$$

and recalling that $P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) = \rho$, then

$$\partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -\rho(1-\rho), \quad (49)$$

$$\partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -\rho(1-\rho) r_{ik}. \quad (50)$$

By substituting Equation (49) and (50) into the truncated Taylor series of Equation (48),

$$P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2}) \approx \rho - \rho(1-\rho)\delta\lambda_{i1} - \rho(1-\rho) r_{ik}\lambda_{i2}. \quad (51)$$

The Lagrange multipliers ($\lambda_{i1}$, $\lambda_{i2}$) are then solved for by substituting Equation (51) into the equations of constraint written in Equations (46b) and (46c).

**Application of approximate distribution to $\Lambda_1$**

Application of Equation (51) to the equation of constraint of Equation (46b),

$$\begin{aligned} \Lambda_1 &= N\rho - \sum_{k=1}^{N} \rho - \rho(1-\rho)\delta\lambda_{i1} - \rho(1-\rho) r_{ik}\lambda_{i2} \\ &= N\rho - N\rho + N\rho(1-\rho)\delta\lambda_{i1} + N\rho(1-\rho)\,\mathbb{E}[R_i]\lambda_{i2}, \end{aligned}$$

where $\mathbb{E}[R_i] = \frac{1}{N}\sum_{k=1}^{N} r_{ik} = (N+1)/2$. By definition $\Lambda_1 = 0$, resulting in

$$\delta\lambda_{i1} = -\,\mathbb{E}[R_i]\lambda_{i2}. \tag{52}$$

**Application of approximate distribution to $\Lambda_2$**

Similarly, for the constraint in Equation (46c),

$$\begin{aligned}
\Lambda_2 &= N\rho\big(\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - \sum_{k=1}^{N} \rho r_{ik} - \rho(1-\rho) r_{ik}\delta\lambda_{i1} - \rho(1-\rho) r_{ik}^2\lambda_{i2} \\
&= N\rho\big(\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - N\rho\,\mathbb{E}[R_i] + N\rho(1-\rho)\,\mathbb{E}[R_i]\delta\lambda_{i1} + N\rho(1-\rho)\,\mathbb{E}[R_i^2]\lambda_{i2}, \\
&= -N\rho(1-\rho)\left[\Delta_i - \mathbb{E}[R_i]\delta\lambda_{i1} - \mathbb{E}[R_i^2]\lambda_{i2}\right],
\end{aligned}$$

where $\mathbb{E}[R_i^2] = \frac{1}{N}\sum_{k=1}^{N} r_{ik}^2$. By definition $\Lambda_2 = 0$, resulting in

$$\Delta_i = \mathbb{E}[R_i]\delta\lambda_{i1} + \mathbb{E}[R_i^2]\lambda_{i2}. \tag{53}$$

**Determining $\lambda_{i1}$, $\lambda_{i2}$, and the approximate maximum entropy distribution**

We first determine $\lambda_{i2}$ by substituting $\delta\lambda_{i1}$ from Equation (52) into Equation (53),

$$\begin{aligned}
\Delta_i &= \left(-\,\mathbb{E}[R_i]^2 + \mathbb{E}[R_i^2]\right)\lambda_{i2} \\
\implies \lambda_{i2} &= \frac{\Delta_i}{\sigma^2}
\end{aligned} \tag{54}$$

where $\sigma^2 = \mathbb{E}[R_i^2] - \mathbb{E}[R_i]^2$ is simply the variance of a uniform discrete random variable between 1 and $N$. Then, by substituting Equation (54) into Equation (52),

$$\lambda_{i1} = \log\left(\frac{1-\rho}{\rho}\right) - \frac{\Delta_i}{\sigma^2}\,\mathbb{E}[R_i] \tag{55}$$

With these expressions for $\lambda_{i1}$ and $\lambda_{i2}$, setting $\sigma^2 = (N^2-1)/12$, and setting $E[R_i] = (N+1)/2$ the first order approximation of the maximum entropy distribution around an uninformative classifier is,

$$P_i(y_k = 1 | r_{ik}) = \left(1 + e^{\log\left(\frac{1-\rho}{\rho}\right) + \frac{12\Delta_i}{N^2-1}\left(r_{ik} - \frac{N+1}{2}\right)}\right)^{-1}.$$

The proof is completed by substituting $N_1/N$ for $\rho$, in which $(1-\rho)/\rho = (N-N_1)/N_1$.

■

## Appendix E. Description of UCI Datasets

We used the following five UCI Datasets in this manuscript: Bank Marketing, Ionosphere, Mammographic Mass, Parkinsons and Yeast datasets. We next briefly describe these datasets.

**Bank Marketing (Moro et al., 2014):**
This data aims to predict whether a client would subscribe to a product (bank term deposit) or not ('no') based on a marketing survey conducted using phone calls. The data has 20 features including age,job,marital status, and education.

**Ionosphere (Sigillito et al., 1989):**
This is a radar dataset that was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere and "Bad" returns are those that do not; their signals pass through the ionosphere.

**Mammographic Masses (Elter et al., 2007):**
This data set is used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006.

**Parkinsons (Little et al., 2007):**
This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each feature corresponds to a particular voice measure, and each sample corresponds to one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

**Yeast (Nakai and Kanehisa, 1991):**
The objective of this data is to use 9 descriptors to predict the localizations (called cellular components) of proteins in a yeast' s cell.