Statistical Tests for Joint Analysis of Performance Measures

Benavoli, A., & de Campos, C. P. (2016). Statistical Tests for Joint Analysis of Performance Measures. In J. Suzuki, & M. Ueno (Eds.), Advanced Methodologies for Bayesian Networks. (pp. 76-92). (Lecture Notes in Computer Science; Vol. 9505). Springer. DOI: 10.1007/978-3-319-28379-1_6

Published in:
Advanced Methodologies for Bayesian Networks

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Statistical Tests for Joint Analysis of Performance Measures

Alessio Benavoli[1] and Cassio P. de Campos[2]

[1] IDSIA, USI, SUPSI**
Manno-Lugano, Switzerland
[2] Queen's University Belfast
Belfast, UK

Abstract. Recently there has been an increasing interest in the development of new methods using Pareto optimality to deal with multi-objective criteria (for example, accuracy and architectural complexity). Once one has learned a model based on their devised method, the problem is then how to compare it with the state of art. In machine learning, algorithms are typically evaluated by comparing their performance on different data sets by means of statistical tests. Unfortunately, the standard tests used for this purpose are not able to jointly consider performance measures. The aim of this paper is to resolve this issue by developing statistical procedures that are able to account for multiple competing measures at the same time. In particular, we develop two tests: a frequentist procedure based on the generalized likelihood-ratio test and a Bayesian procedure based on a multinomial-Dirichlet conjugate model. We further extend them by discovering conditional independences among measures to reduce the number of parameter of such models, as usually the number of studied cases is very reduced in such comparisons. Real data from a comparison among general purpose classifiers is used to show a practical application of our tests.

1 Introduction

In many real applications of machine learning, we often need to consider the trade-off between multiple conflicting objectives. For instance, measures like accuracy and architectural complexity are clearly two different (possibly conflicting) criteria. This issue can be tackled by considering a multi-objective decision making approach.

There are two main approaches to multi-objective decision making. The weighted-sum approach, which consists of transforming the original multi-objective problem into a single-objective problem by using a weighted formula; The Pareto approach, which considers directly the original multi-objective problem and searches for non-dominated solutions, that is, solutions that are not worse than any other solution with respect to all criteria.

In a weighted-sum approach, a multi-objective problem is transformed into a single-objective problem by a numerical weight function that is assigned to objectives and

** Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), Scuola universitaria professionale della Svizzera italiana (SUPSI), Università della Svizzera italiana (USI)

then values of the weighted criteria are combined into a single value according to the weights. One of the reasons for its popularity is its simplicity. However, there are several drawbacks associated to it. First, the definition of weights in these formulas is often ad-hoc or requires great domain knowledge which might not be available. Second, the optimal solution strongly depends on that particular weight function, which misses the opportunity to find other models that might be actually more interesting to the user, for instance, representing a better trade-off between different criteria. Third, a weighted formula involving a linear combination of different criteria is meaningless in many scenarios, as the criteria may be non-commensurable (comparison of apples and oranges).

In the Pareto approach, instead of transforming a multi-objective problem into a single-objective problem and then solving it by using a single-objective decision making, a multi-objective algorithm is used to solve the original multi-objective problem. The advantage of the Pareto approach is that it can cope with any kind of non-commensurable criteria. Recently there has been an increasing interest in the development of new learning methods able to cope simultaneously with multi-objective criteria using Pareto optimality [1,2,3,4]. The disadvantage comes from the *power* of the Pareto approach in situations where a good weight function can be devised, as the Pareto approach is more conservative than using the weighted-sum idea. In this work we assume that a good weight function is not available. Consider for instance the work in [3], where it is proposed a multi-objective Pareto based optimization method for simultaneous optimization of architectural complexity and accuracy for Polynomial Neural Networks (PNN). By using multiple data sets, they compare their method with the state-of-art method for learning PNN, producing the results presented in Table 1.

	New		State of art	
	Accuracy	Complexity	Accuracy	Complexity
IRIS	97.8	38.4	95.3	50.0
WINE	98.3	26.9	92.3	24.0
PIMA	72.1	28.6	65.3	37.7
BUPA	70.3	23.4	69.1	36.0

Table 1: Architectural complexity and accuracy of two learning methods for PNN [3].

Based on Table 1, [3] claims that a multi-objective approach (jointly optimizing architectural complexity and accuracy) is clearly beneficial. Can we say that their method is clearly better than the state of art for both criteria and also for each of them independently? For which criterion is it superior (respectively inferior)? To answer these questions we need a method that statistically assesses whether an algorithm is better than another in terms of all criteria. To the best knowledge of the authors, this method is lacking in machine learning and so it could not be used in [3].

Competing methods/algorithms are typically compared by means of a statistical test, whose aim is to assess whether an algorithm is significantly better than another (statistically comparing their performance on different data sets or problem instances). For comparing two algorithms over a collection of data sets, the most common approaches

are the sign test or the Wilcoxon signed-rank test [5], however these tests are only able to cope with one performance measure (criterion) at a time, that is, they cannot consider a multi-objective approach without resorting to the weighted-sum approach described earlier. In this paper, we develop two tests that are able to cope jointly with multiple performance measures without having to somehow combine them: a frequentist procedure based on the generalized likelihood-ratio test and a Bayesian procedure based on a multinomial-Dirichlet conjugate model. We further extend them by discovering conditional independences among measures to reduce the number of parameters of such models, an important add-on since usually the number of data sets on which methods are compared is reduced. Applications of these new tests are numerous. Here we use real data from a comparison of general purpose classification methods to show a clear practical application of the tests.

2 Joint Analysis of Performance criteria

Let $M_1, \ldots, M_m$ be a set of m performance measures (criteria) and assume that we are going to compare two algorithms A and B by jointly using these measures.

Definition 1. *We call a 'dominance statement' for B against A a sequence of m dominance conditions:*

$$D^{(BA)} = [\succ, \succ, \prec, \ldots, \succ]\,,$$

where the comparison $\succ$ (or $\prec$) in the i-th entry of the vector $D^{(BA)}$ means that algorithm B is better than A (respectively, A is better than B) on measure M_i. ■

Our goal is to make inferences on *dominance statements* by evaluating the m performance measures for the algorithms A and B on n different case studies (for instance, data sets, problem instances, etc). In other words, we want to decide which $D^{(BA)}$ is the most appropriate for A and B given tables with values $M_{ij}^{(\mathrm{Alg})}$ representing the j-th measure for the algorithm $\mathrm{Alg} \in \{A, B\}$ in the i-th case study:

$$\mathbf{M}^{(\mathrm{Alg})} = \begin{bmatrix} M_{11}^{(\mathrm{Alg})} & M_{12}^{(\mathrm{Alg})} & \ldots & M_{1m}^{(\mathrm{Alg})} \\ M_{21}^{(\mathrm{Alg})} & M_{22}^{(\mathrm{Alg})} & \ldots & M_{2m}^{(\mathrm{Alg})} \\ \vdots & \vdots & \vdots & \vdots \\ M_{n1}^{(\mathrm{Alg})} & M_{n2}^{(\mathrm{Alg})} & \ldots & M_{nm}^{(\mathrm{Alg})} \end{bmatrix}. \tag{1}$$

Given the matrix of performances $\mathbf{M}^{(A)}$ and $\mathbf{M}^{(B)}$, we first build the binary matrix $\mathbf{X} = [\mathbf{M}^{(B)} \succ \mathbf{M}^{(A)}]$, whose entry x_{ij} is equal to one if algorithm B is better than algorithm A for the j-th measure in the i-th case study and zero otherwise. We assume that ties do not exist.[3] To each matrix $\mathbf{X}$ we associate a count vector $\mathbf{n}$, whose entries

[3] If there are ties we treat a tie in a measure by a standard approach: we replicate the case with it into two and divide the weight of such case by two (this process might need to be performed multiple times until no ties are present in the data). Such approach preserves the sample size and fairly allocates ties between the algorithms being compared.

represent the counts for each one of the 2^m possible *dominance statements* (many of which might be zero).

Example 1. Consider the comparison of two algorithms in terms of accuracies M_1 (expressed in percent values in the first row) and time M_2 (in seconds, shown in the second row) on 12 data sets:

$$\begin{aligned}\mathbf{M}^A &= \begin{bmatrix}85 & 87 & 87 & 91 & 91 & 91 & 94 & 94 & 94 & 94 & 94 & 94\\ 8 & 11 & 11 & 12 & 12 & 12 & 16 & 16 & 16 & 16 & 16 & 16\end{bmatrix}^T,\\ \mathbf{M}^B &= \begin{bmatrix}84 & 86 & 86 & 92 & 92 & 92 & 95 & 95 & 95 & 95 & 95 & 95\\ 9 & 10 & 10 & 13 & 13 & 13 & 15 & 15 & 15 & 15 & 15 & 15\end{bmatrix}^T\end{aligned} \tag{2}$$

where ^{T} denotes transpose.

The matrix $\mathbf{X} = [\mathbf{M}^{(B)} \succ \mathbf{M}^{(A)}]$ is:[4]

$$\mathbf{X} = \begin{bmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T. \tag{3}$$

Hence, we derive that the dominance statement $[\prec, \prec]$ (or $[0, 0]$), which means that B is worse than A on both measures, is observed $n_0 = 1$ time; the statement $[\prec, \succ]$ (or $[0, 1]$), which means that B is worse than A on the first measure but better on the second, is observed $n_1 = 2$ times; the statement $[\succ, \prec]$ (or $[1, 0]$) is observed $n_2 = 3$ times; the statement $[\succ, \succ]$ (or $[1, 1]$) is observed $n_3 = 6$ times. Hence, we have that $\mathbf{n} = [1, 2, 3, 6]$ (a binary order is used for the entries of $\mathbf{n}$). ■

The matrix $\mathbf{X}$ or, equivalently, the vector $\mathbf{n}$, include all the information that we will use to derive our tests. While this approach might seem to lose information because we only account for the sign of each difference $M_{ij}^{(\mathrm{Alg})} - M_{ij}^{(\mathrm{Alg}')}$, there is no effective way of using the actual value of the difference across multiple measures if these measures are assumed to be expressed in *incomparable units*, as in this case no procedure could be used to compare the measures jointly or to collapse the measures into a single one in order to run standard tests (using some weighting function; we assume that normalizing the measures is not an option either, as it entails an additional assumption about the measures which might not hold). On the other hand, the sign of the difference is a proper comparable value among measures regardless of the particular meaning of each of them. In fact, we point out that the measures $M_{ij}^{(\mathrm{Alg})}$ can themselves be obtained from any arbitrary procedure (including statistical tests), as we only assume that the sign of the difference $M_{ij}^{(\mathrm{Alg})} - M_{ij}^{(\mathrm{Alg}')}$ is available (and we properly account for ties). This provides us with a very general setting, allowing for numerous applications.

3 Generalized Likelihood Ratio Test

We derive a simple null hypothesis significance test for the joint analysis of performance measures. We denote by θ_k, for $k = 0, \ldots, 2^m - 1$, the probability of obtaining one of

[4] An algorithm is better ($\succ$) than another when it has higher accuracy and lower computational time.

the 2^m possible *dominance statements*. Hence, $\theta_k \geq 0$ and $\sum_{k=0}^{2^m-1} \theta_k = 1$. We have enumerated the *dominance statements* according to their "binary order", so that θ_0 is the probability of the statement $[\prec, \ldots, \prec, \prec]$, θ_1 is the probability of $[\prec, \ldots, \prec, \succ]$, θ_2 is the probability of $[\prec, \ldots, \prec, \succ, \prec]$, etc. Our goal is to find if there is a statement that is significantly more likely than all others based on the observation matrix $\mathbf{X}$. It is clear that $\mathbf{n}$ is a sufficient statistic for this test, since its k-th entry n_k corresponds to the counts for the k-th statement. Hence, to achieve our goal, we can perform a *Generalized Likelihood Ratio Test* (GLRT):

$$\lambda(\mathbf{n}) = \frac{\max_{\boldsymbol{\theta} \in \Theta^*} L(\boldsymbol{\theta}|\mathbf{n})}{\max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}|\mathbf{n})}, \text{ where } L(\boldsymbol{\theta}|\mathbf{n}) = \prod_{k=0}^{2^m-1} \theta_k^{n_k}, \tag{4}$$

$\boldsymbol{\theta} = [\theta_0, \ldots, \theta_{2^m-1}]$, Θ is the simplex for $\boldsymbol{\theta}$, $\Theta^* = \{\boldsymbol{\theta} \in \Theta : \ \theta_{i^*} \leq \max(\boldsymbol{\theta} \setminus \theta_{i^*})\}$ (we abuse notation and indicate by $\boldsymbol{\theta} \setminus \theta_{i^*}$ all thetas apart from θ_{i^*}) and $i^* = \operatorname{argmax}_{i=0,\ldots,2^m-1} n_i$. The rationality behind Eq.(4) is that we are testing two hypothesis: $(H_0)\ \theta_{i^*} \leq \max(\boldsymbol{\theta} \setminus \theta_{i^*})$ and $(H_1)\ \theta_{i^*} > \max(\boldsymbol{\theta} \setminus \theta_{i^*})$. Under H_0, the value of $\boldsymbol{\theta}$ which better explains the observations is the maximum likelihood estimate (MLE) subject to the constraint that $\boldsymbol{\theta} \in \Theta^*$. Its likelihood is the numerator of Eq. (4). The value of $\boldsymbol{\theta}$ which maximizes the likelihood is instead the MLE subject to $\boldsymbol{\theta} \in \Theta$. It is clear that $0 \leq \lambda(\mathbf{n}) \leq 1$. GLRT employs $\lambda(\mathbf{n})$ as a test statistic and rejects H_0 for small values of $\lambda(\mathbf{n})$, that is, when $\lambda(\mathbf{n}) \leq \rho$, where the value of ρ is determined by fixing the type-I error to be α. By Wilks' theorem, for large n, $-2\log(\lambda(\mathbf{n}))$ is chi-square distributed with one degree of freedom [6,7]. Hence, the rejection zone for the null hypothesis is approximately equal to

$$\mathcal{R} = \left\{\mathbf{n} : \ -2\log(\lambda(\mathbf{n})) > \chi^2_{1,\alpha}\right\}, \tag{5}$$

where α is the confidence level. Therefore, to apply GLRT, we must only compute $\lambda(\mathbf{n})$.

Theorem 1. *Given the count vector* $\mathbf{n}$*, it holds that*

$$\lambda(\mathbf{n}) = \frac{\left(\frac{n_a+n_b}{2}\right)^{n_a+n_b}}{n_a^{n_a} n_b^{n_b}}, \tag{6}$$

where n_a *is the greatest value among* $n_0, \ldots, n_{2^m-1}$ *and* n_b *the second greatest.* ■

Proof. The maximum likelihood estimate of $\boldsymbol{\theta}$ subject to the constraint $\boldsymbol{\theta} \in \Theta$ is

$$\left(\frac{n_0}{n}, \frac{n_1}{n}, \ldots, \frac{n_{2^m-1}}{n}\right),$$

in fact the only constraint on $\boldsymbol{\theta}$ in this case is that its elements sum up to 1. The maximum likelihood estimate of $\boldsymbol{\theta}$ subject to the constraint $\Theta^* = \{\boldsymbol{\theta} \in \Theta : \ \theta_{i^*} \leq \max(\boldsymbol{\theta} \setminus \theta_{i^*})\}$ can be computed using KKT conditions of optimality for optimization problems subject to inequality constraints. To obtain this estimate let us assume without

loss of generality that $n_0 \geq n_1 \geq n_2 ...$ Note that $i^* = \text{argmax}_{i=0,\dots,2^m-1} n_i$ and so considering the equality constraint $\theta_{i^*} = \max(\boldsymbol{\theta} \setminus \theta_{i^*})$, we have that the maximum likelihood estimate of $\boldsymbol{\theta}$ is

$$\left(\frac{n_c}{n}, \frac{n_c}{n}, \frac{n_2}{n}, \dots, \frac{n_{2^m-1}}{n}\right),$$

where $n_c = (n_0 + n_1)/2$. Then the likelihood ratio is

$$\frac{(\frac{n_c}{n})^{n_0} \cdot (\frac{n_c}{n})^{n_1} \dots (\frac{n_{2^m-1}}{n})^{n_0}}{(\frac{n_0}{n})^{n_0} \cdot (\frac{n_1}{n})^{n_1} \dots (\frac{n_{2^m-1}}{n})^{n_0}} = \frac{n_c^{n_0+n_1}}{n_0^{n_0} n_1^{n_1}},$$

which proves the theorem. ■

In case $n_a = n_b$, we have $\lambda(\mathbf{n}) = 1$ and $-2\log(\lambda(\mathbf{n})) = 0$, so that the null hypothesis can never be rejected. It can be shown that:

Theorem 2. *The GLRT (Eq.* (5)*) is (asymptotically) calibrated (i.e., it controls the Type-I error) for a prescribed significance level α obtaining the maximum type-I error when $n_a + n_b = n$.* ■

This can be proven using an approach similar to that described in [8, Ex. 21.2].

Example 2. In Example 1, $m = 2$ and Eq.(2) *yields $L(\boldsymbol{\theta}|\mathbf{n}) = \theta_0\theta_1^2\theta_2^3\theta_3^6$, where θ_0 is the probability of the statement $[\prec, \prec]$, θ_1 of $[\prec, \succ]$, θ_2 of $[\succ, \prec]$ and θ_3 of $[\succ, \succ]$. Hence, $n_a = 6$, $n_b = 3$, the statistic $\lambda(\mathbf{n}) = \frac{(\frac{9}{2})^9}{3^3 6^6} \approx 0.6$ and the p-value is 0.313. Given the value of the p-value, we cannot conclude that B is better than A on both performance measures.* ■

GLRTs have the disadvantage that they do not provide the probability of the hypotheses, but only its p-value under H_0. This means that we do not have any information about the probability of the alternative hypothesis being true. To address this issue, in the next section we propose a Bayesian hypothesis test for testing a certain *dominance statement*.

4 Bayesian test

We implement the Bayesian hypothesis test by following a Bayesian estimation approach, that is, by estimating the posterior probability of the vector of parameters $\boldsymbol{\theta}$. Given the count vector $\mathbf{n}$, the likelihood of $\boldsymbol{\theta}$ given the data is given by the right-hand side of Eq. (4), which is a multinomial distribution. As prior we then consider a Dirichlet distribution: $p(\boldsymbol{\theta}) \propto \prod_{k=0}^{2^m-1} \theta_k^{\alpha_k - 1}$, where $\alpha_k > 0$ are the parameters of the Dirichlet distribution. In the rest of the paper, we will always use the symmetric prior $\alpha_k = 1/2^m$ (however, we can also use other priors such as the Jeffreys prior $\alpha_k = \frac{1}{2}$, or some robust prior model [9]). By conjugacy, the posterior is also a Dirichlet with updated

parameters $n_k + \alpha_k$. In the Bayesian setting, to make inferences on a *dominance statement*, we have simply to compute the posterior probabilities $P(\theta_i > \max(\boldsymbol{\theta} \setminus \theta_i)|\mathbf{n})$, for $i = 0, \ldots, 2^m - 1$. This is the posterior probability that θ_i (associated to the i-statement) is greater than all other $\theta_{\neg i}$ values.

Proposition 1. *It holds that* $\sum_{i=0}^{2^m-1} P(\theta_i > \max(\boldsymbol{\theta} \setminus \theta_i)|\mathbf{n}) = 1.$ ■

This result follows from the simple fact that $P(\theta_i = \theta_j|\mathbf{n}) = 0$ (i.e., since θ_i are continuous variables, it is clear that $P(\theta_i = \theta_j|\mathbf{n}) = 0$ since any probability density function on continuous variables assign probability zero to singletons). Hence, the above posterior probabilities enclose all the available information on the *dominance statements*. These probabilities can easily be computed by Monte Carlo sampling on the space of vectors $\boldsymbol{\theta}$ from the posterior Dirichlet distribution and then by counting the fraction of times we see $\theta_i > \max(\boldsymbol{\theta} \setminus \theta_i)$, for every i.

Example 3. Take again Example 1. We already know that $L(\boldsymbol{\theta}|\mathbf{n}) = \theta_0\theta_1^2\theta_2^3\theta_3^6$*, where* θ_0 *is the probability of the statement* $[\prec, \prec]$*,* θ_1 *of* $[\prec, \succ]$*,* θ_2 *of* $[\succ, \prec]$ *and* θ_3 *of* $[\succ, \succ]$*. The posterior probabilities of hypotheses are:* $P(\theta_0 > \theta_{\neg 0}|\mathbf{n}) \approx 0.013$*,* $P(\theta_1 > \theta_{\neg 1}|\mathbf{n}) \approx 0.051$*,* $P(\theta_2 > \theta_{\neg 2}|\mathbf{n}) \approx 0.136$*, and* $P(\theta_3 > \theta_{\neg 3}|\mathbf{n}) \approx 0.80$*. Hence the most probable dominance statement is* $[\succ, \succ]$ *and its probability is* 0.8*. These probabilities have been computed by Monte Carlo sampling as discussed above.* ■

5 Bayesian Network

The columns of $\mathbf{X} = [\mathbf{M}^{(B)} \succ \mathbf{M}^{(A)}]$ can be seen as binary random variables $\mathcal{M} = \{M_1, \ldots, M_m\}$ representing which algorithm is better according to that measure. Because of possible stochastic conditional independences between these variables, the estimation of a joint probability $p(\mathcal{M})$ can be improved by using a Bayesian network (BN). A BN can be defined as a triple $(\mathcal{G}, \mathcal{M}, \mathcal{P})$, where $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ is a directed acyclic graph (DAG) with $V_\mathcal{G}$ a collection of m nodes associated to the random variables $\mathcal{M}$ (a node per variable), and $E_\mathcal{G}$ a collection of arcs; $\mathcal{P}$ is a collection of conditional probabilities $p(M_i|PA_i)$ where PA_i denotes the parents of M_i in the graph (PA_i may be empty), corresponding to the relations of $E_\mathcal{G}$. In a Bayesian network, the Markov condition states that every variable is conditionally independent of its non-descendants given its parents. This structure induces a joint probability distribution by the factorization $p(M_1, \ldots, M_m) = \prod_i p(M_i|PA_i)$. Let $\boldsymbol{\theta}$ be the entire vector of parameters such that $\theta_{ijk} = p(M_i = k|PA_i = j)$, where $k \in \{0, 1\}$, $j \in \{1, ..., 2^{|PA_i|}\}$ and $i \in \{1, \ldots, m\}$. Note that this represents a different parametrization with respect to the $\boldsymbol{\theta}$ of previous sections, but a simple transformation can be used to compute those values through the factorization expression. Given the table $\mathbf{X}$ with m measures and n case studies, the structure learning problem in Bayesian networks is to find a DAG $\mathcal{G}$ that maximizes its posterior probability, that is, $\mathcal{G}^* = \text{argmax}_{\mathcal{G} \in \boldsymbol{\mathcal{G}}}\, p(\mathcal{G}|\mathbf{X})$, with $\boldsymbol{\mathcal{G}}$ the set of all DAGs

over node set $\mathcal{M}$.

$$p(\mathcal{G}|\mathbf{X}) \propto p(\mathcal{G}) \cdot \int p(\mathbf{X}|\mathcal{G},\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta},$$

where $p(\boldsymbol{\theta}|\mathcal{G})$ is the prior of $\boldsymbol{\theta}$ for a given graph $\mathcal{G}$, assumed to be a symmetric Dirichlet with positive hyper-parameter α^*:

$$p(\boldsymbol{\theta}|\mathcal{G}) = \prod_{i=1}^{m} \prod_{j=1}^{2^{|PA_i|}} \Gamma(\frac{\alpha^*}{2^{|PA_i|}}) \prod_{k=0}^{1} \frac{\theta_{ijk}^{\frac{\alpha^*}{2^{|PA_i|+1}}-1}}{\Gamma(\frac{\alpha^*}{2^{|PA_i|+1}})}. \tag{7}$$

α^* is usually referred to as the Equivalent Sample Size (ESS). Such computation is known as the Bayesian Dirichlet Equivalent Uniform (BDeu) criterion [10,11], where we assume parameter independence and modularity [12]. We also assume $\alpha^* = 1$ and that there is no preference for any graph and set $p(\mathcal{G})$ as uniform.

In order to find the graph representing the best set of conditional independences over the space of all possible DAGs $\boldsymbol{\mathcal{G}}$, multiple approaches have been proposed in the literature. Because the number of measures is hardly above 15 to 20 and they are all binary, the combination of properties of the BDeu score [13] with a dynamic programming algorithm [14] usually suffices. Otherwise one might use more sophisticated ideas [15,16,17], which can deal with a greater number of variables. Given the optimal graph $\mathcal{G}$, we can employ the discovered conditional independences to write the joint distribution for $\mathcal{M}$ opportunely:

$$p(\mathbf{X}|\mathcal{G},\boldsymbol{\theta}) = \prod_{i=1}^{m} \prod_{j=1}^{2^{|PA_i|}} \theta_{ij0}^{n_{ij0}} (1-\theta_{ij0})^{n_{ij1}},$$

where n_{ijk} counts the number of times $(M_i = k \wedge PA_i = j)$ in the data. Combined with the prior $p(\boldsymbol{\theta}|\mathcal{G})$ of Eq. (7), this can be used to compute $P(\theta_i > \max(\boldsymbol{\theta} \setminus \theta_i)|\mathbf{X})$ by Monte Carlo sampling as before (even if different from previous sections, the parametrization of $\boldsymbol{\theta}$ used here also works for that). The advantages of using Bayesian networks are as follows. First, by using the $p(\mathcal{G}|\mathbf{X})$, the dependence model underlying the distribution is automatically adapted to what can be inferred from data, and so one usually needs fewer observations to learn a good model than when working with the full joint. Second, the graph can be used to identify relations between measures and how they are associated, which can be for instance used to ignore measures that are not able to help in discriminating the algorithms. Third, computations can be carried out efficiently (at least when we restrict ourselves to a couple of tens of variables). We will illustrate these benefits later on.

6 Experiments

In this section, we apply our tests to compare seven classifiers on 80 data sets (10 runs of 10-folds cross-validation) and using several performance measures. We have considered the following classifiers 'AODE' (C1), 'Bayes net' (C2), 'Bayes.NaiveBayes'

(C3), 'trees.J48graft' (C4), 'trees.RandomForest' (C5), 'trees.bagging' (C6) and 'logistic' (C7). We have performed all the experiments using WEKA [18], which implements all such classifiers, and analyzed the results using our `R` package. We note that our purpose is not to conclude in favor or against any of the classifiers, but to illustrate the use of our new approaches to compare them.

6.1 Accuracy and FPR-TPR

In this experiment, we have considered three measures. Accuracy is the percentage of correct predictions of a model, the most common measure to evaluate a classifier. For a binary classification problem, the true positive rate (*TPR*) defines how many correct positive results occur among all positive samples available during the test. The false positive rate (*FPR*), on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. It is well known that accuracy is highly dependent on TPR and FPR (in the binary case it is just a convex combination of them). We compare the classifiers using (i) only accuracy and (ii) FPR-TPR jointly, and expect to see a great agreement between the results of (i) and (ii) because of the strong dependence between those measures. For (i) we use the Wilcoxon sign-rank test (which has more power than the sign test), and our tests for (ii). Matrix (8) (left) reports the statistical comparison of the seven classifiers performed by considering accuracy only. The numerical values in the matrix are the p-values of Wilcoxon sign-rank test computed on the direction ($\prec$ or $\succ$) corresponding to the highest value of the statistic (most likely direction to refute the null hypothesis). For instance, the meaning of the first matrix entry is as follows: C_1 has been found better than C_2 with p-value close to zero. Conversely, the first element in the second row means that C_2 has been found worse than C_3 (but non-significant with p-value 0.46). All pairwise comparisons with p-values less than $\alpha/2$ (e.g, $\alpha = 0.1$ or 0.05) are significant. To control the family-wise type-I error of many pairwise comparisons, the significance level should be adjusted by the Bonferroni correction (or other more efficient approaches) [5]. Hereafter, we report the p-values of the frequentist tests, so the implementation of such corrections is straightforward.

$$
\begin{array}{c}
\\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succ 0 & \succ 0 & \prec & \prec .17 & \succ 0 & \succ 0 \\
 & \prec .46 & \prec 0 & \prec .046 & \succ 0 & \succ 0 \\
 & & \prec 0 & \prec .048 & \succ 0 & \succ 0 \\
 & & & \succ .026 & \succ 0 & \succ 0 \\
 & & & & \succ 0 & \succ 0 \\
 & & & & & \prec 0
\end{pmatrix}
\end{array}
\quad
\begin{array}{c}
\\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succsim .99 & \succsim .99 & \precsim .99 & \precsim .85 & \precsim 1 & \precsim 1 \\
 & \succsim .27 & \precsim 1 & \precsim .92 & \succsim 1 & \succsim 1 \\
 & & \precsim 1 & \precsim .90 & \succsim 1 & \succsim 1 \\
 & & & \succsim .93 & \succsim 1 & \succsim 1 \\
 & & & & \succsim 1 & \succsim 1 \\
 & & & & & \precsim 1
\end{pmatrix}
\end{array}
\tag{8}
$$

Matrix (8) (right) reports the comparison performed with the Bayesian test considering jointly TPR and -FPR (negative FPR so that as higher as better). In this case, each entry of the matrix represents the most probable joint dominance and the numerical value the relative probability. The first element says that C_1 is jointly better than

C_2 because has higher -FPR (so lower FPR) and higher TPR and this statement holds with posterior probability 0.99. The test using the Bayesian network model achieved almost equal results for the probabilities (variations only because of Monte Carlo, data not shown), because the two measures are well correlated (so the Bayesian network inferred the joint model as the most probable, which reduced it to the standard Bayesian test without the Bayesian network). Also the GLRT is consistent with the results obtained by the Bayesian test. For instance, its p-values relative to the C_1 row are $0.014, 0.024, 0, 0.29, 0, 0$. Apart from 0.29 all the p-values are significant for $\alpha = 0.05$. A reason to prefer GLRT to the Bayesian test is that we have shown that it is calibrated to type-I error. On the other hand the probabilities returned by the Bayesian test have a more direct interpretation. For this reason, in the following we will just show the results of the Bayesian test. Comparing the two matrices is clear that the results are quite in agreement (smaller p-values correspond to higher probabilities and vice versa). The advantage of approach with multiple measures is that it is able to jointly consider them and thus its conclusions have additional meaning.

6.2 Accuracy, F-measure and Weighted-AUC

In this section we compare the classifiers using accuracy, F-measure and weighted-AUC: (i) separately; (ii) considering pairwise combinations of these measures; (iii) considering the three measures together.

For the case of Accuracy and Weighted-AUC, Matrix (9) (on the left) reports the results of the comparison obtained considering separately each of these measures (each cell contains the result for Accuracy on top and Weighted-AUC below it), while Matrix (9) (on the right) is the result of the Bayesian joint test. For performing the separate tests, we have used the Wilcoxon sign-rank test [5]. The numerical values in the Matrix (9) (on the left) are the p-values of Wilcoxon sign-rank test computed on the direction ($\prec$ or $\succ$) corresponding to the highest value of the statistic (most likely direction to refute the null hypothesis). For instance, the meaning of the comparison C_1 versus C_5, is as follows: C_1 has been found worse than C_5 in accuracy (with p-value 0.17) and better in Weighted-AUC (with p-value 0.14). All pairwise comparisons with p-values less than $\alpha/2$ (e.g, $\alpha = 0.1$ or 0.05) are significant.[5] Matrix (9) (on the right) reports the comparison performed with the Bayesian test considering jointly Accuracy and Weighted-AUC. In this case, each entry of the matrix represents the most probable joint dominance statement and the numerical value is the relative probability. Comparing the two matrices, there are two cases where the tests are in clear contradiction (in bold) and a case (C_4 vs. C_7), where the joint comparison gives an evident advantage in power. This means that C_4 is better than C_7 jointly on both accuracy and Weighted-AUC, while this is not true when the two performance measures are considered separately. Therefore, it is evident that decisions derived by a joint test can be very different from the decisions carried out using a separate test for each performance measure. If the

[5] To control the family-wise type-I error of many pairwise comparisons, the significance level should be adjusted by the Bonferroni correction (or other more efficient approaches) [5]. Hereafter, we report the p-values of the frequentist tests, so the implementation of such corrections is straightforward.

goal is to compare algorithms considering jointly the measures, then it is more appropriate to use the new methods proposed here. The GLRT is overall consistent with the results obtained by the Bayesian test (results not shown). For instance, its p-value for "C_4 better than C_7 on both the performance measures" is almost zero (so "very" significant). One reason to prefer GLRT to the Bayesian test is that we have shown that it is calibrated to type-I error. On the other hand, the probabilities returned by the Bayesian test have a more direct interpretation. The choice depends on the user's needs.

$$
\begin{array}{c}
\\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succ 0 \\ \succ 0 & \succ 0 \\ \succ 0 & \succ 0 & \prec 0 \\ \succ 0 & \succ 0 & \succ 0 & \prec .17 \\ \succ 0 & \succ 0 & \succ 0 & \succ .14 & \succ 0 \\ \succ 0 & \succ 0 & \succ 0 & \succ 0 & \succ 0 & \succ 0
\end{pmatrix}
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succ .99 & \succ .96 & \succ .99 & \prec .64 & \succ 1 & \succ 1 \\
 & \prec .42 & \succ .99 & \prec \mathbf{.72} & \succ 1 & \succ 1 \\
 & & \succ .99 & \prec \mathbf{.72} & \succ 1 & \succ 1 \\
 & & & \prec .56 & \succ 1 & \succ 1 \\
 & & & & \succ 1 & \succ 1 \\
 & & & & & \prec 1
\end{pmatrix}
\end{array}
. \qquad (9)
$$

Now we consider Weighted-AUC and F-measure together. Matrix (10) (on the left) reports the results of the comparison based on separate tests (each cell contains the result for Weighted-AUC on top and F-measure below it), while Matrix (10) (on the right) regards the Bayesian joint test. There are five cases where the tests are in contradiction (in bold). In particular, in the comparisons C_2 vs. C_5 and C_3 vs. C_5, the Bayesian test asserts that C_5 is jointly better with probability 0.91, while the separate tests do not find a significant dominance. Again for C_4 vs. C_7, it is evident that the joint comparison gives an advantage in power.

$$
\begin{array}{c}
\\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succ 0 \\ \succ 0 & \succ 0 \\ \succ 0 & \succ 0 & \prec 0 \\ \succ 0 & \succ 0 & \succ 0 & \prec \mathbf{0} \\ \succ 0 & \succ 0 & \succ 0 & \succ \mathbf{.14} & \succ 0 \\ \succ 0 & \succ 0 & \succ 0 & \succ 0 & \succ 0 & \succ 0
\end{pmatrix}
\end{array}
\begin{array}{c}
\begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\
\begin{pmatrix}
\succ .99 & \succ .99 & \prec 1 & \prec \mathbf{.81} & \succ 1 & \succ 1 \\
 & \prec \mathbf{.37} & \prec 1 & \prec \mathbf{.91} & \succ 1 & \succ 1 \\
 & & \prec 1 & \prec \mathbf{.91} & \succ 1 & \succ 1 \\
 & & & \prec \mathbf{.55} & \succ 1 & \succ 1 \\
 & & & & \succ 1 & \succ 1 \\
 & & & & & \prec 1
\end{pmatrix}
\end{array}
. \qquad (10)
$$

Finally we consider the three performance measures together. Matrix (11) reports the result of the Bayesian joint test.

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{array} \begin{array}{c} \begin{array}{cccccc} C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \end{array} \\ \begin{pmatrix} .99 & .99 & 1 & \mathbf{.81} & 1 & 1 \\ & \mathbf{.31} & 1 & \mathbf{.91} & 1 & 1 \\ & & 1 & \mathbf{.91} & 1 & 1 \\ & & & \mathbf{.55} & 1 & 1 \\ & & & & 1 & 1 \\ & & & & & 1 \end{pmatrix} \end{array}. \tag{11}$$

We can then assert that C_1 is better than C_2 and C_3 jointly on all performance measures. Overall, C_5 appears to be jointly the best classifier followed by C_4. By using the Bayesian network inference to compare C_4 and C_5, we achieve the very same conclusions (results not shown). The interesting outcome of that inference is that we can graphically see the relation between measures in Figure 1, which is automatically learned from the matrix of measures, and not surprisingly, all three measures of classification accuracy are dependent.

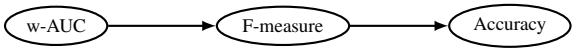


Fig. 1: Three measures used to compare C_4 and C_5 and their (in)dependences.

6.3 Comparison Using Six Measures

In this section we compare the same seven classifiers but now using six performance measures jointly: Accuracy, F-measure, weighted-AUC, Kappa statistics, root mean squared error (RMSE), and mean absolute error (MAE). In order to illustrate the capabilities of the proposed approach, let us take on the task of comparing the classifiers C_1 and C_2. By using the BN and the learned conditional (in)dependences displayed in Figure 2, we obtain the probability of C_1 to be better than C_2 jointly in all six measures to be 0.5, while the value reaches 0.9 without using the BN, which suggests that an unreliable decision could be taken because independent measures where assumed to be dependent (the model without the Bayesian network was learned with very few data, about 80 cases for a parameter space of dimension 63, which is clearly insufficient). From Figure 2 we see that RMSE and MAE are independent measures with respect to the others and each other. With such information, we can look to their importance separately. Using the Bayesian test for MAE we get a very low probability of 0.54 towards C_2, while RMSE achieves 0.99 towards C_2. The other four connected measures

in Figure 2 achieve probability 0.9999 towards C_1. Hence we are able to identify the source of this difference between the result with the Bayesian network and without it, which clarifies the measures under which one classifier is better than the other. Further applications are numerous, but they go beyond the scope of this work.

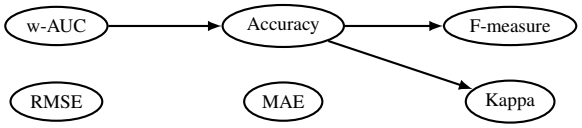
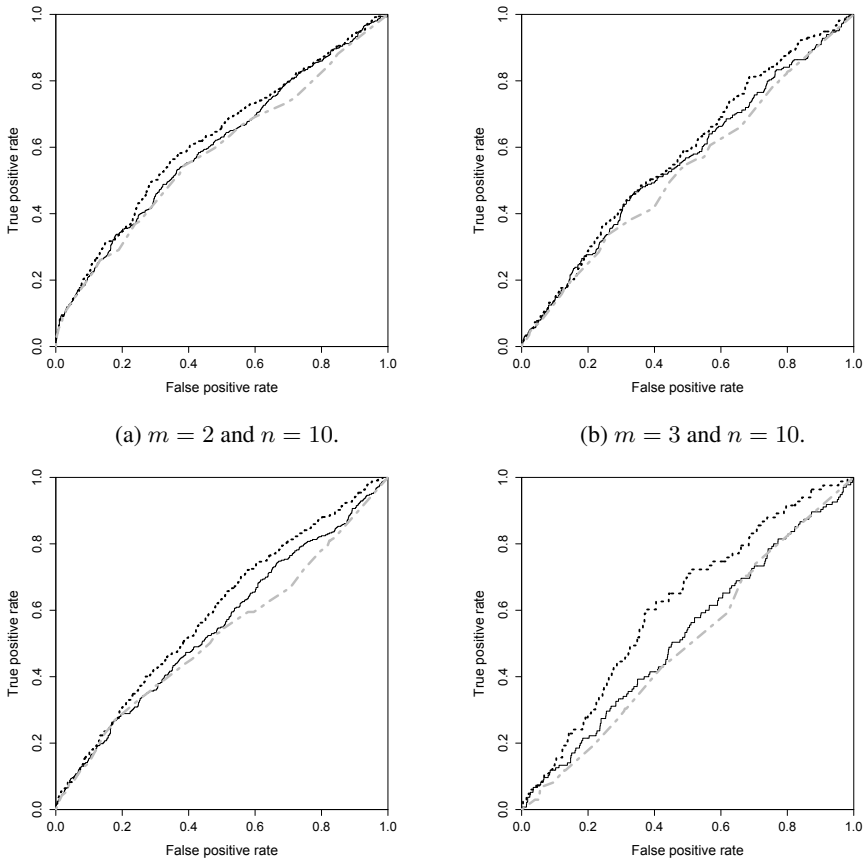


Fig. 2: Six measures used to compare classifiers C_1 and C_2 and their dependences.

6.4 Simulation study

Finally, we perform a simulated study to understand the benefit of using the Bayesian networks. We study scenarios with m equal to 2, 3 and 5 measures from which we randomly draw the multinomial parameters, that is, $2^2 - 1 = 3$, $2^3 - 1 = 7$ and $2^5 - 1 = 31$ independent parameters, respectively. We label each test case as follows: if the maximum θ is greater than the second greatest plus 0.1%, then this is labelled as a case where there is a difference between the maximum and the others. Otherwise we say the maximum is not greater than the others (and we force the maximum and second greatest to be equal to each other). Then we randomly generate n samples (n =10, 20 or 50) from the distribution and run the GLRT and the Bayesian test with and without the support of the Bayesian network to learn the underlying distribution from data. For each test case, we record the probability that the maximum parameter is greater than the others (or the p-value in the case of the GLRT). This procedure is repeated one thousand times for cases where the maximum is greater (so *positive* cases) and one thousand times with the maximum equal to the second greatest value (*negative* cases). The results over these two thousand test cases are used to build a receiver operating characteristic (ROC) curve according to the usual procedure: True/false positive/negative are defined by varying the threshold for the probability (or respectively the p-value) such that the method takes a decision of whether it is a positive or negative case. In this way, we obtain the percentage (over two thousand test cases) of true positive, true negative, false positive and false negative for each method for each threshold. The curves with the GLRT (gray dashed-dotted) and the Bayesian test with the Bayesian network (black dashed) and without it (black contiguous) are shown in Figure 3 for different values of m and n. In all cases, the GLRT is equal or inferior to the Bayesian test, and the Bayes test with the Bayesian network version is always equal or superior to the Bayesian test alone.

We repeat the experiment but we now assume that the five measures are independent from each other. In this scenario we expect the method with the Bayesian network to be superior, as it can estimate a more appropriate joint distribution (given the limited amount of data). Again we randomly draw the parameters of the multinomial (respect-



(a) $m = 2$ and $n = 10$.

(b) $m = 3$ and $n = 10$.

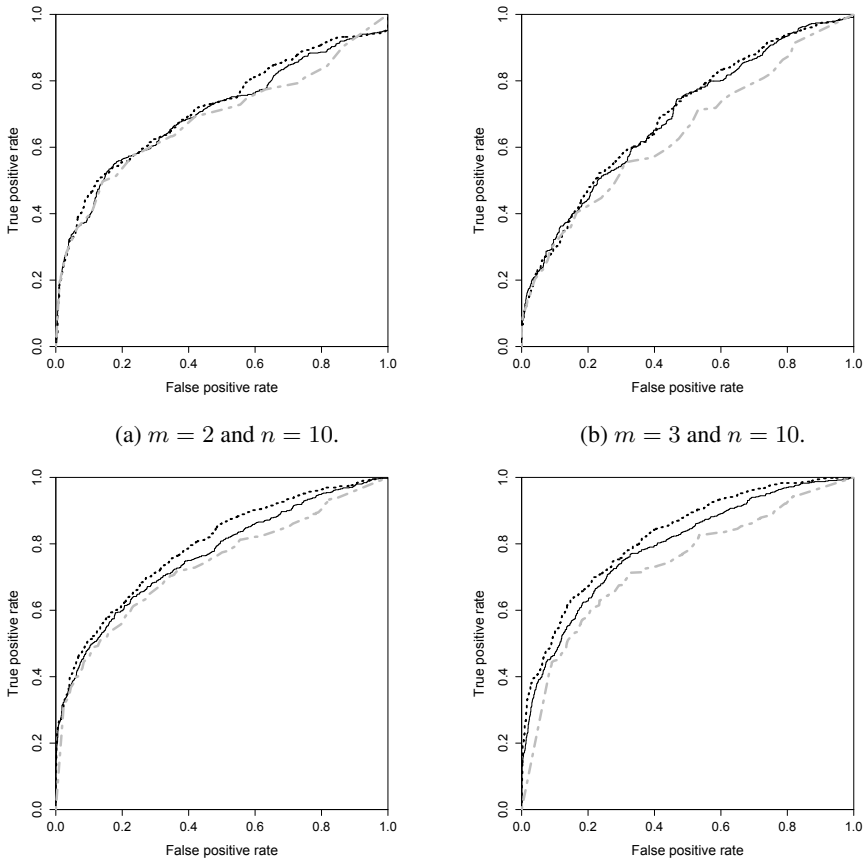
(c) $m = 3$ and $n = 20$.

(d) $m = 5$ and $n = 50$.

Fig. 3: ROC curves for the GLRT (gray dashed-dotted) and the Bayesian test with (black dashed) and without (black contigous) the Bayesian network use during learning. Distributions and data (n samples) are generated for a domain with m measures uniformly at random.

ing the independence assumption), then the data and we label the cases as before. The ROC curves for this scenario are shown in Figure 4.

The area under the curves of each method and each scenario is presented in Table 2. The values obtained by GLRT are always inferior to those of the Bayesian test. The latter has consistently produced better results with the support of the Bayesian network for learning the distribution. The superiority of the method with the Bayesian network is justified by the better estimation of the joint distribution with its underlying independence assessments.



(a) $m = 2$ and $n = 10$.

(b) $m = 3$ and $n = 10$.

(c) $m = 3$ and $n = 20$.

(d) $m = 5$ and $n = 50$.

Fig. 4: ROC curves for the GLRT (gray dashed-dotted) and the Bayesian test with (black dashed) and without (black contigous) the Bayesian network use during learning. Distributions and data (n samples) are generated for a domain with m measures uniformly at random assuming that all measures are independent from each other.

7 Conclusions

In machine learning and artificial intelligence, a very important task is to compare the performance of algorithms on different case studies and to use multiple different performance measures. This is typically performed using statistical tests. In this paper, we have developed new statistical tests that are able to compare the algorithms considering all the performance measures jointly. This allows for example to make statements such as a classifier is jointly better than another on multiple measures as well as on particular subsets of measures, which can be identified with the use of a Bayesian network modelling the (in)dependences among measures. With artificial and real-data examples

m	n	Type	GLRT	Bayesian test	Bayesian test + BN
2	10	indep	0.686	0.703	0.715
2	10	full	0.583	0.601	0.622
3	10	indep	0.641	0.688	0.694
3	10	full	0.530	0.555	0.577
3	20	indep	0.735	0.764	0.791
3	20	full	0.524	0.549	0.590
5	50	indep	0.735	0.790	0.822
5	50	full	0.500	0.522	0.613

Table 2: Area under the ROC curve for each of the methods in each analyzed scenario. m is the number of measures, n the number of data points over which the measures are compared, and Type describes whether the simulation sampled the parameters without restriction (full) or with the forced assumption that each measure is independent of each other (indep).

we have shown that the decisions derived by a joint test can be very different from the decisions carried out using a separate test for each performance measure. We argue that the ideas developed here can offer a new way for comparing algorithms using multiple performance measures. Future work includes the exploration of applications and the further use of the Bayesian network structure to understand the relations between performance measures and their importance for the evaluation of algorithms. Moreover, we plan to extend this approach to be able to compare multiple measure on multiple algorithms at the same time.

References

1. S. Dehuri and S.-B. Cho, "Multi-criterion pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art," *Computer Science Review*, vol. 3, no. 1, pp. 19–40, 2009.
2. W. Cai, S. Chen, and D. Zhang, "A multiobjective simultaneous learning framework for clustering and classification," *Neural Networks, IEEE Transactions on*, vol. 21, no. 2, pp. 185–200, 2010.
3. C. Shi, X. Kong, S. Y. Philip, and B. Wang, "Multi-objective multi-label classification.," in *SDM*, pp. 355–366, SIAM, 2012.
4. K. jen Hsiao, K. Xu, J. Calder, and A. O. Hero, "Multi-criteria anomaly detection using pareto depth analysis," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 845–853, Curran Associates, Inc., 2012.
5. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
6. S. S. Wilks, "The large-sample distribution of the likelihood ratio for testing composite hypotheses," *Ann. Math. Statist.*, vol. 9, pp. 60–62, 03 1938.
7. J. Rice, *Mathematical statistics and data analysis*. Cengage Learning, 2006.
8. A. DasGupta, *Asymptotic theory of statistics and probability*. Springer (NY), 2008.
9. P. Walley, "Inferences from multinomial data: learning about a bag of marbles," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 3–57, 1996.

10. W. Buntine, "Theory refinement on Bayesian networks," in *UAI-92* (B. D. D'Ambrosio, P. Smets, and P. P. Bonissone, eds.), (San Francisco, CA), pp. 52–60, Morgan Kaufmann, 1991.
11. G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.
12. D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.
13. C. P. de Campos and Q. Ji, "Properties of Bayesian Dirichlet scores to learn Bayesian network structures," in *AAAI Conference on Artificial Intelligence*, pp. 431–436, AAAI Press, 2010.
14. T. Silander and P. Myllymaki, "A simple approach for finding the globally optimal bayesian network structure," in *22nd Conference on Uncertainty in Artificial Intelligence*, (Arlington, Virginia), pp. 445–452, AUAI Press, 2006.
15. M. Barlett and J. Cussens, "Advances in Bayesian network learning using integer programming," in *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, UAI'13, pp. 182–191, 2013.
16. C. P. de Campos and Q. Ji, "Efficient structure learning of Bayesian networks using constraints," *Journal of Machine Learning Research*, vol. 12, pp. 663–689, Mar 2011.
17. C. Yuan and B. Malone, "Learning optimal Bayesian networks: A shortest path perspective," *Journal of Artificial Intelligence Research*, vol. 48, pp. 23–65, 2013.
18. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.