New Gener. Comput. (2017) 35:69–86
DOI 10.1007/s00354-016-0005-8

SPECIAL FEATURE

Joint Analysis of Multiple Algorithms and Performance Measures

Cassio P. de Campos[1] · **Alessio Benavoli**[2]

Received: 15 April 2016 / Accepted: 15 August 2016 / Published online: 12 December 2016

Abstract 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 time complexity). Once one has developed an approach to a problem of interest, 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. Standard tests used for this purpose are able to consider jointly neither performance measures nor multiple competitors at once. The aim of this paper is to resolve these issues by developing statistical procedures that are able to account for multiple competing measures at the same time and to compare multiple algorithms altogether. 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 parameters of such models, as usually the number of studied cases is very reduced in such comparisons. Data from a comparison among general purpose classifiers are used to show a practical application of our tests.

✉ Alessio Benavoli
alessio@idsia.ch

Cassio P. de Campos
c.decampos@qub.ac.uk

[1] Queen's University, Belfast, UK

[2] IDSIA, Manno, Switzerland

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 dealing with multi-objective decision making. The weighted sum approach, which consists of transforming the original multi-objective problem into a single-objective problem 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 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 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 [1–4]. 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], which proposes a multi-objective Pareto-based optimization method for simultaneous optimization of architectural complexity and accuracy for Polynomial Neural Networks (PNN). Using multiple data sets, they compare their method with the state-of-art method for learning PNN, producing the results presented in Table 1.

Based on Table 1, the authors [3] claim 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

Table 1 Architectural complexity and accuracy of two learning methods for PNN [3] Higher values are better

	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

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 limited. Applications of these new tests are numerous. Here, we use data from a comparison of general purpose classification methods to show a clear practical application of the tests.

This work is an extension of the paper presented at AMBN 2015 [6]. We generalize those ideas to deal with the comparison of multiple algorithms at once, that is, we design new statistical tests that can compare multiple algorithms under multiple performance measures. The idea is to label each ordering of the algorithms into a category in our parameter space, so we can identify which is the most probable ordering and whether such most probable ordering is significantly more probable than the others.

The paper is divided as follows. Section 2 defines the dominance statement that we later use to design the statistical tests. Sections 3 and 4 present, respectively, our frequentist and Bayesian statistical tests for dealing with multiple measures jointly. Section 5 shows how to use Bayesian networks to improve the estimation of the joint distributions and thus increase the quality of the results of the tests. Section 6 describes how the designed tests can be easily adapted to deal with multiple algorithms and multiple measures jointly. Section 7 describes and presents our experiments with synthetic data, and finally Sect. 8 concludes the work and points out directions for future research.

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 ith 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 jth measure for the algorithm $\mathrm{Alg} \in \{A, B\}$ in the ith 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 jth measure in the ith case study and zero otherwise. We assume that ties do not exist.[1] To each matrix $\mathbf{X}$, we associate a count vector $\mathbf{n}$, whose entries 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:[2]

[1] 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.

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

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

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 lexicographic order is used for the entries of $\mathbf{n}$). □

The matrix $\mathbf{X}$ or, equivalently, the vector $\mathbf{n}$, includes 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 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.

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 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 kth entry n_k corresponds to the counts for the kth statement. Hence, to achieve our goal, we can perform a *Generalized Likelihood Ratio Test* (GLRT):

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

$\boldsymbol{\theta} = [\theta_0, \ldots, \theta_{2^m-1}]$, Θ is the simplex for $\boldsymbol{\theta}$, $\Theta^* = \{\boldsymbol{\theta} \in \Theta : \theta_{i^*} \leq \max(\boldsymbol{\theta} \backslash \theta_{i^*})\}$ (we abuse notation and indicate by $\boldsymbol{\theta} \backslash \theta_{i^*}$ all thetas apart from θ_{i^*}) and $i^* = \text{argmax}_{i=0,\ldots,2^m-1} n_i$. The rationality behind Eq. (4) is that we are testing two hypothesis: (H_0) $\theta_{i^*} \leq \max(\boldsymbol{\theta} \backslash \theta_{i^*})$ and (H_1) $\theta_{i^*} > \max(\boldsymbol{\theta} \backslash \theta_{i^*})$. 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 [7, 8]. 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 $\Theta^* = \{\boldsymbol{\theta} \in \Theta : \theta_{i^*} \leq \max(\boldsymbol{\theta} \backslash \theta_{i^*})\}$ can be computed using KKT conditions of optimality for optimization problems subject to inequality constraints [9]. To obtain this estimate let us assume without loss of generality that $n_0 \geq n_1 \geq n_2 \cdots$, and so the constraint is $\theta_0 \leq \theta_1$. To facilitate the derivation, let us work with the log-likelihood function (it has the same maximum). The KKT conditions are

$$\frac{n_0}{\theta_0} = \mu + \nu \quad \text{and} \quad \frac{\mathrm{n}_1}{\theta_1} = \mu - \nu \quad \text{and} \quad \forall \mathrm{k} \leq 2 : \frac{\mathrm{n}_\mathrm{k}}{\theta_\mathrm{k}} = \mu \quad \text{and} \quad \nu \cdot (\theta_0 - \theta_1) = 0.$$

Multiplying each one by θ_k and summing them we obtain $n = \mu + \nu \cdot (\theta_0 - \theta_1) = \mu$, and the maximum happens when $\theta_0 = \theta_1 = \frac{n_0+n_1}{2n}$. So we have that the maximum likelihood estimate of $\boldsymbol{\theta}$ is

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

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

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

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 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 [10, 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 $\lambda(\mathbf{n}) = \frac{\binom{9}{2}^9}{3^3 6^6} \approx 0.6$ 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 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*. On the other hand, GLRT is extremely fast when compared to the Bayesian test.

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 are 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 [11]). 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 to simply compute the posterior probabilities $P(\theta_i > \max(\boldsymbol{\theta}\backslash\theta_i)|\mathbf{n})$, for $i = 0,\ldots,2^m-1$. This is the posterior probability that θ_i (associated with the i-statement) is greater than all other $\theta_{\neg i}$ values. □

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

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 assigns probability zero to singletons). Hence, the above posterior probabilities consider 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 $\theta_i > \max(\boldsymbol{\theta} \backslash \theta_i)$, 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. □

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 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 $\mathcal{M}$.

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

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^*:

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

α^* is usually referred to as the Equivalent Sample Size (ESS). Such computation is known as the Bayesian–Dirichlet Equivalent Uniform (BDeu) criterion [12, 13], where we assume parameter independence and modularity [14]. We also assume $\alpha^* = 1$ and that there is no preference for any graph and set $p(\mathcal{G})$ as uniform.

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 15–20 and they are all binary, the combination of properties of the BDeu score [15] with a dynamic programming algorithm [16] usually suffices. Otherwise, one might use more sophisticated ideas [17–19], 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:

$$p(\mathbf{X}|\mathcal{G}, \boldsymbol{\theta}) = \prod_{i=1}^{m} \prod_{j=1}^{2^{|PA_i|}} \theta_{ij0}^{n_{ij0}} (1 - \theta_{ij0})^{n_{ij1}}, \tag{8}$$

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 $P(\theta_i > \max(\boldsymbol{\theta} \backslash \theta_i)|\mathbf{X})$ 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, 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, i.e., performance measures). We will illustrate these benefits later on.

Comparing Multiple Algorithms

The results so far dealt with the comparison between two algorithms under multiple performance measures. In this section, we generalize such approach to compare multiple algorithms. To do so, we must define an extended dominance statement, where each element corresponds to an ordering of the goodness of the desired algorithms. Let $M_1, \ldots, M_m$ be a set of m performance measures (criteria) and assume that we are going to compare l algorithms $A_1, \ldots, A_l$ by jointly using these measures.

Definition 2 We call a 'dominance statement' for algorithms $A_1, \ldots, A_l$ a sequence of m dominance conditions:

$$D^{(A_1,\ldots,A_l)} = [o_1, o_2, o_3, \ldots, o_m],$$

where the value o_i in the ith entry of the vector $D^{(A_1,\ldots,A_l)}$ is an integer from 0 to

$l! - 1$ indicating a permutation of the algorithms $A_1, \ldots, A_l$ which respects their performances according to measure M_i. For simplicity, we sort the $l!$ possible permutations in lexicographical order and assign an integer accordingly. □

In short, the elements o_i of the vector $D^{(A_1,\ldots,A_l)}$ tell us what is the order of goodness (best to worst) among the algorithms being compared for that measure. Ties are dealt as explained before. Using this definition, the matrix $\mathbf{X}$ defined earlier can now be built directly with the values from $D^{(A_1,\ldots,A_l)}$, which become the rows of $\mathbf{X}$ (one row for each dataset). Everything proceeds as before, but now the state space for the generalized likelihood ratio test and for the Bayesian test are not anymore of size 2^m but $(l!)^m$ instead. Hence, $\theta_k \geq 0$ and $\sum_{k=0}^{(l!)^m-1} \theta_k = 1$, and there is an obvious correspondence between each θ_k and the order which it represents (this can be inferred from the value k). All definitions and derivations in Sects. 3, 4 and 5 can be easily adapted by employing this extended space of parameters. For instance, Expressions (7) and (8) need to account for variables of the Bayesian network taking on $(l!)$ values instead of two. Moreover, an interesting property of this extension to multiple algorithms is that the marginal models (that is, projecting the joint distribution for multiple algorithms into any two particular algorithms) yield exactly the same results as the simplified formulation created for only two algorithms, so this idea generalizes that version described earlier in a sound manner. For the sake of simplicity, we will not present the results of Sects. 3, 4 and 5 again, since they are not particularly depending on the number of categories in the state space, and hence they trivially work for this new situation presented here with multiple algorithms.

Experiments

We perform a simulated study to understand the benefit of using the Bayesian networks. We study scenarios with m equal to 2 and 3 measures and with 3 algorithms (that is, $l = 3$) from which we uniformly draw at random the multinomial parameters, that is, $(3!)^2 - 1 = 35$ and $(3!)^3 - 1 = 215$ independent parameters, respectively. This reflects a scenario of full dependence between measures (with probability 1). We label each test case (that is, each draw) as follows: if the maximum θ is greater than the second greatest plus 0.1%, then this is labeled 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 =50, 100 or 200) 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

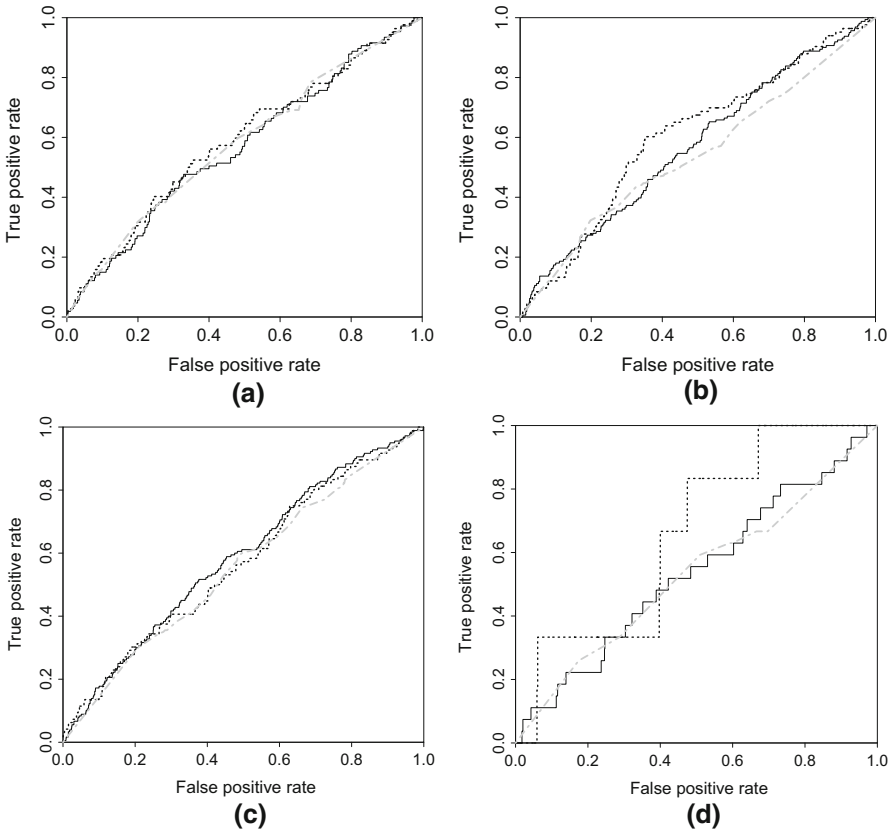

Fig. 1 ROC curves for the GLRT (*gray dashed-dotted*) and the Bayesian test with (*black dashed*) and without (*black contiguous*) the Bayesian network use during learning. Distributions and data (n samples) are generated for a domain with m measures

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 Fig. 1 for different values of m and n. In general, the GLRT is equal or inferior to the Bayesian test, and the Bayes test with the Bayesian network version is usually superior to the Bayesian test alone. We notice that in some cases the curves are barely better than random guess (which would correspond to the identity function in the ROC graph). This happens because there are too many parameters to learn in the multinomial with respect to the amount of data. We see that with the increase of data ($n = 200$) and independent

measures the curve begins to improve with respect to random guesses, because the true models are simpler and hence need fewer data to be learned.

We repeat the experiment but we now assume that the measures are independent of each other. In this scenario, we expect the method with the Bayesian network to be superior than using the full joint distribution, as it can estimate a more appropriate distribution (given the limited amount of data). The idea is that the Bayesian network can learn the fact that the measures are independent (this fact is not disclosed to the methods, as in practice we usually would not know it beforehand). Again we uniformly draw at random the parameters of the multinomial (respecting the independence assumption among all measures), then we draw the data and we label the cases as before. The ROC curves for this scenario are shown in Fig. 2. Again, the Bayesian test with the Bayesian network achieves the best curves.

Table 2 shows the area under the curves for each method and scenario when comparing the three simulated algorithms. The values obtained by GLRT are inferior to those of the Bayesian test. The latter has consistently produced better

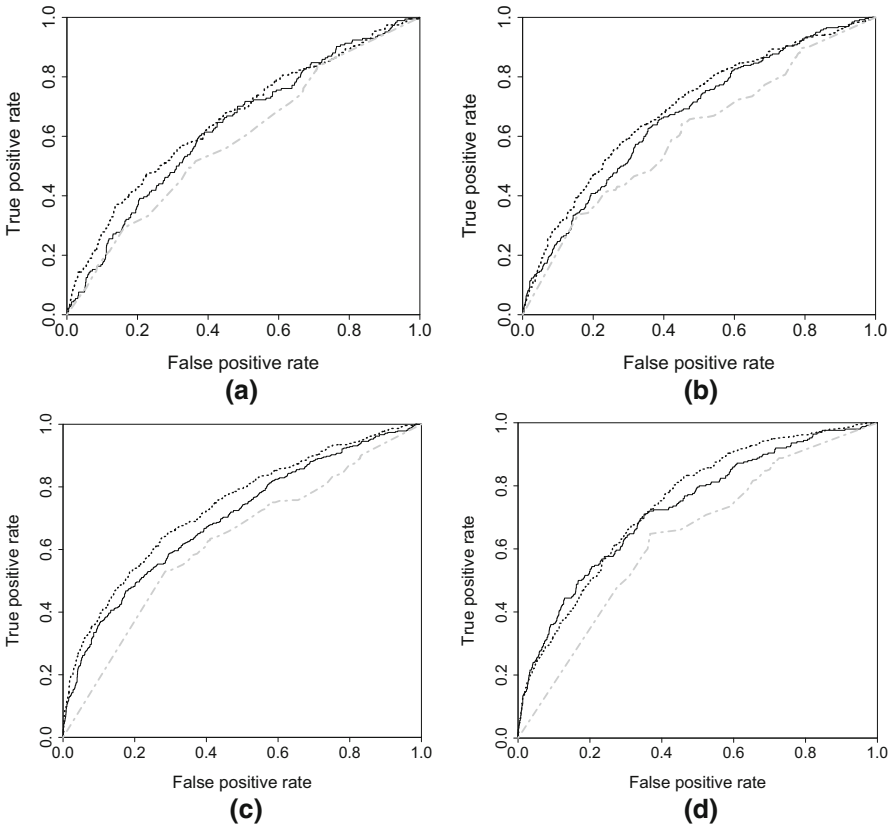

Fig. 2 ROC *curves* for the GLRT (*gray dashed-dotted*) and the Bayesian test with (*black dashed*) and without (*black contiguous*) the Bayesian network use during learning. Distributions and data (*n* samples) are generated for a domain with *m* measures uniformly at random assuming that all measures are independent from each other

Table 2 Area under the ROC curve for each method in each scenario when comparing 3 algorithms

m	*n*	Type	GLRT	Bayesian test	Bayesian test + BN
2	50	Indep	0.588	0.632	0.657
2	50	Full	0.578	0.567	0.586
2	100	Indep	0.607	0.667	0.694
2	100	Full	0.541	0.569	0.603
2	200	Indep	0.624	0.697	0.735
2	200	Full	0.555	0.585	0.567
3	100	Indep	0.592	0.671	0.688
3	100	Full	0.490	0.588	0.517
3	300	Indep	0.636	0.728	0.743
3	300	Full	0.530	0.536	0.656

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)

Table 3 Area under the ROC curve for each method in each scenario when comparing 2 algorithms

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

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)

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. We notice in Table 2 that the Bayesian network version loses to one of the others when data were generated from the full distribution and the amount of data is not so small. The reasoning is that, in these cases, the data were enough to fit well the joint distribution directly without the need of the Bayesian network.

We repeat the experiment with only two algorithms, and record the area under the ROC curve as before. Results are presented in Table 3. With only two algorithms the results are even more clear in favor of the Bayesian test with the support of the Bayesian network.

Data of Classification Problems

In this section, we apply our tests to compare classifiers on 80 data sets (10 runs of tenfold cross-validation) and using several performance measures. We start by comparing classifiers two by two, and later we demonstrate the case of more than two classifiers being compared at once. We have considered the following classifiers 'AODE' (C_1), 'Bayes net' (C_2), 'Bayes.NaiveBayes' (C_3), 'trees.J48graft' (C_4), 'trees.RandomForest' (C_5) and 'logistic' (C_6). We have performed all the experiments using WEKA [20], which implements all such classifiers, and analyzed the results using simple scripts in R. 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. The data and measures used in the analysis are available at http://www.cs.qub.ac.uk/∼c.decampos/ngc2016/. As illustration, we compare the classifiers using accuracy, F-measure and weighted-AUC: (1) separately; (2) considering pairwise combinations of these measures; (3) 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 signed-rank test [5]. The numerical values in the Matrix (9) (on the left) are the p values of Wilcoxon signed-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.[3] 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_6) where the joint comparison gives an evident advantage in power. This means that C_4 is better than C_6 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 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_6 on both the performance measures" is almost zero (so "very" significant). The choice between GLRT and the Bayesian test depends on the user's needs.

[3] To control the family-wise type I error of many pairwise comparisons, the significance level should be adjusted, as previously described.

$$
\begin{array}{c}
\begin{array}{ccccc} C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \\
\left(\begin{array}{ccccc}
\begin{array}{c}\succ 0\\ \succ 0\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec .17\\ \succ .14\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & \begin{array}{c}\prec .46\\ \prec .37\end{array} & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec \mathbf{.05}\\ \succ \mathbf{.47}\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec \mathbf{.05}\\ \succ \mathbf{.43}\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & & & \begin{array}{c}\succ .026\\ \prec 0\end{array} & \begin{array}{c}\succ 0\\ \succ .14\end{array} \\
 & & & & \begin{array}{c}\succ 0\\ \succ 0\end{array}
\end{array}\right)
\begin{array}{c} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{array}
\left(\begin{array}{ccccc}
\begin{array}{c}\succ\\ \succ\end{array} .99 & \begin{array}{c}\succ\\ \succ\end{array} .96 & \begin{array}{c}\prec\\ \succ\end{array} .99 & \begin{array}{c}\prec\\ \prec\end{array} .64 & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & \begin{array}{c}\succ\\ \prec\end{array} .42 & \begin{array}{c}\prec\\ \succ\end{array} .99 & \begin{array}{c}\prec\\ \prec\end{array} \mathbf{.72} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & \begin{array}{c}\succ\\ \succ\end{array} .99 & \begin{array}{c}\prec\\ \prec\end{array} \mathbf{.72} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & & \begin{array}{c}\succ\\ \prec\end{array} .56 & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & & & \begin{array}{c}\succ\\ \succ\end{array} 1
\end{array}\right).
\end{array}
\tag{9}
$$

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_6, it is evident that the joint comparison gives an advantage in power.

$$
\begin{array}{c}
\begin{array}{ccccc} C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \\
\left(\begin{array}{ccccc}
\begin{array}{c}\succ 0\\ \succ 0\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec \mathbf{0}\\ \succ \mathbf{.14}\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & \begin{array}{c}\succ \mathbf{.27}\\ \prec \mathbf{.37}\end{array} & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec \mathbf{0}\\ \succ \mathbf{.47}\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & & \begin{array}{c}\prec 0\\ \succ 0\end{array} & \begin{array}{c}\prec \mathbf{0}\\ \succ \mathbf{.43}\end{array} & \begin{array}{c}\succ 0\\ \succ 0\end{array} \\
 & & & \begin{array}{c}\succ \mathbf{0}\\ \prec \mathbf{0}\end{array} & \begin{array}{c}\succ 0\\ \succ .14\end{array} \\
 & & & & \begin{array}{c}\succ 0\\ \succ 0\end{array}
\end{array}\right)
\begin{array}{c} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{array}
\left(\begin{array}{ccccc}
\begin{array}{c}\succ\\ \succ\end{array} .99 & \begin{array}{c}\succ\\ \succ\end{array} .99 & \begin{array}{c}\prec\\ \succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\end{array} \mathbf{.81} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & \begin{array}{c}\succ\\ \prec\end{array} \mathbf{.37} & \begin{array}{c}\prec\\ \succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\end{array} \mathbf{.91} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & \begin{array}{c}\prec\\ \succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\end{array} \mathbf{.91} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & & \begin{array}{c}\succ\\ \prec\end{array} \mathbf{.55} & \begin{array}{c}\succ\\ \succ\end{array} 1 \\
 & & & & \begin{array}{c}\succ\\ \succ\end{array} 1
\end{array}\right).
\end{array}
\tag{10}
$$

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

$$
\begin{array}{c}
\begin{array}{ccccc} C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \\
\begin{array}{c} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{array}
\left(\begin{array}{ccccc}
\begin{array}{c}\succ\\ \succ\succ\end{array} .99 & \begin{array}{c}\succ\\ \succ\succ\end{array} .99 & \begin{array}{c}\prec\\ \succ\succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\succ\end{array} \mathbf{.81} & \begin{array}{c}\succ\\ \succ\succ\end{array} 1 \\
 & \begin{array}{c}\succ\\ \prec\prec\end{array} \mathbf{.31} & \begin{array}{c}\prec\\ \succ\succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\succ\end{array} \mathbf{.91} & \begin{array}{c}\succ\\ \succ\succ\end{array} 1 \\
 & & \begin{array}{c}\prec\\ \succ\succ\end{array} 1 & \begin{array}{c}\prec\\ \prec\succ\end{array} \mathbf{.91} & \begin{array}{c}\succ\\ \succ\succ\end{array} 1 \\
 & & & \begin{array}{c}\succ\\ \prec\prec\end{array} \mathbf{.55} & \begin{array}{c}\succ\\ \succ\succ\end{array} 1 \\
 & & & & \begin{array}{c}\succ\\ \succ\succ\end{array} 1
\end{array}\right).
\end{array}
\tag{11}
$$

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. 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

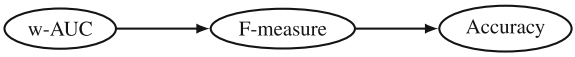

Fig. 3 Three measures used to compare C_4 and C_5 and their (in)dependences

that we can graphically see the relation between measures in Fig. 3, which is automatically learned from the matrix of measures, and not surprisingly, all three measures of classification accuracy are dependent.

Finally, we have run the joint tests using all three measures and classifiers C_1, C_2, C_3 all at once, as described in Sect. 6. GLRT and the Bayesian test with and without the Bayesian network have all concluded the following orderings: for accuracy and F-measure, we have $C_1 \succ C_2 \succ C_3$, while for weighted-AUC we have $C_1 \succ C_3 \succ C_2$. These orderings are the same that we have found when performing the pairwise analysis, as can be seen in the first entries of Matrix (11). This result confirms the previous analysis and makes it much stronger, since it states that those are the most probable orderings among the classifiers when joint analysis of measures and classifiers was conducted. Because C_2 is better than C_3 over two measures but worse than it over another, in a Pareto sense, we could say that C_1 is the best classifier, but C_2 and C_3 are not dominant over each other. This kind of situation demonstrates the importance of performing the correct analysis for the problem at hand. Such decision is nevertheless dependent on the objectives of the study and should be taken case by case, since no unique methodology fits all problems.

Conclusions

Comparing algorithms under multiple measures is typically performed using independent statistical tests. In this paper, we have developed new statistical tests that are able to compare multiple algorithms at once and consider all the performance measures jointly. This allows us, 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. These subsets of measures can be identified with the use of a Bayesian network, i.e., by modelling the (in)dependences among measures. With artificial data examples 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. A clear drawback is that we cannot compare too many algorithms at the same time, because the complexity grows exponentially in the number of algorithms being compared. As future work, we intend to overcome this issue by learning an appropriate space of ordering instead of considering all possible ordering among the algorithms (which is factorial in the number of algorithms). We also intend to explore 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.

References

1. Dehuri, S., Cho, S.-B.: Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art. Comput. Sci. Rev. **3**(1), 19–40 (2009)
2. Cai, W., Chen, S., Zhang, D.: A multiobjective simultaneous learning framework for clustering and classification. Neural Netw. IEEE Trans. **21**(2), 185–200 (2010)
3. Shi, C., Kong, X., Yu, P.S., Wang, B.: Multi-Objective Multi-Label Classification. In: Proceedings of the 2012 SIAM International Conference on Data Mining, pp. 355–366. SIAM (2012)
4. Hsiao, K.J., Xu, K., Calder, J., Hero, A.O.: Multi-criteria anomaly detection using pareto depth analysis. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 845–853. Curran Associates Inc, USA (2012)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
6. Benavoli, A., de Campos, C.P.: Statistical Tests for Joint Analysis of Performance Measures, pp. 76–92. Springer International Publishing, Cham (2015)
7. Wilks, S.S.: The large-sample distribution of the likelihood ratio for testing composite hypotheses. Ann. Math. Stat. **9**, 60–62 (1938)
8. Rice, J.: Mathematical statistics and data analysis. Cengage Learning, USA (2006)
9. de Campos, C.P., Tong, Y., Ji, Q.: Constrainted maximum likelihood bayesian network for facial expression recognition. In: European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, vol. 5304, pp. 168–181 (2008)
10. DasGupta, A.: Asymptotic theory of statistics and probability. Springer, New York (2008)
11. Walley, P.: Inferences from multinomial data: learning about a bag of marbles. J. R. Stat. Soc. B **58**(1), 3–57 (1996)
12. Buntine, W.: Theory refinement on Bayesian networks. In: D'Ambrosio, B.D., Smets, P., Bonissone, P.P. (eds.) UAI-92, pp. 52–60. Morgan Kaufmann, San Francisco, CA (1991)
13. Cooper, G.F., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Mach. Learn. **9**, 309–347 (1992)
14. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Mach. Learn. **20**, 197–243 (1995)
15. de Campos, C.P., Ji, Q.: Properties of Bayesian Dirichlet scores to learn Bayesian network structures. In: AAAI Conference on Artificial Intelligence, pp. 431–436. AAAI Press, USA (2010)
16. Silander, T., Myllymaki, P.: A simple approach for finding the globally optimal Bayesian network structure. 22nd Conference on Uncertainty in Artificial Intelligence. Arlington, Virginia, pp. 445–452. AUAI Press, USA (2006)
17. Barlett, M., Cussens, J.: 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)
18. de Campos, C.P., Ji, Q.: Efficient structure learning of Bayesian networks using constraints. J. Mach. Learn. Res. **12**, 663–689 (2011)
19. Yuan, C., Malone, B.: Learning optimal Bayesian networks: A shortest path perspective. J. Artif. Intell. Res. **48**, 23–65 (2013)
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)

Alessio Benavoli received all his degrees in Computer and Control Engineering from the University of Firenze, Italy (Ph.D in 2008 and MS degree in 2004). In 2008, he worked for the international company SELEX-Sistemi Integrati as system analyst. Currently, he is working as Senior researcher at the Dalle Molle Institute for Artificial Intelligence in Lugano (CH). His research interests are in the areas of

imprecise probabilities, Bayesian nonparametrics and control. He has co-authored about 70 peer-reviewed publications.

Cassio P. de Campos is a Reader with Queen's University Belfast, UK. He obtained his Ph.D in 2006 with a thesis about algorithms and computational complexity in Bayesian and credal networks and his Habilitation in 2013 related to the same topics, both from the University of Sao Paulo, Brazil. He works with probabilistic graphical models, imprecise probability, computational complexity and bioinformatics. He serves as committee member of multiple conferences in machine learning and artificial intelligence, and is currently an at-large member in the executive committee of the Society for Imprecise Probability.