An Interval-based Multiobjective Approach to Feature Subset Selection Using Joint Modeling of Objectives and Variables

Hossein Karshenas, Pedro Larrañaga, Qingfu Zhang, and Concha Bielza

***Abstract*—This paper studies feature subset selection in classification using a multiobjective estimation of distribution algorithm. We consider six functions, namely area under ROC curve, sensitivity, specificity, precision, F1 measure and Brier score, for evaluation of feature subsets and as the objectives of the problem. One of the characteristics of these objective functions is the existence of noise in their values that should be appropriately handled during optimization. Our proposed algorithm consists of two major techniques which are specially designed for the feature subset selection problem. The first one is a solution ranking method based on interval values to handle the noise in the objectives of this problem. The second one is a model estimation method for learning a joint probabilistic model of objectives and variables which is used to generate new solutions and advance through the search space. To simplify model estimation, ℓ_1 regularized regression is used to select a subset of problem variables before model learning. The proposed algorithm is compared with a well-known ranking method for interval-valued objectives and a standard multiobjective genetic algorithm. Particularly, the effects of the two new techniques are experimentally investigated. The experimental results show that the proposed algorithm is able to obtain comparable or better performance on the tested datasets.**

***Index Terms*—Feature subset selection, Multiobjective optimization, Estimation of distribution algorithm, Joint objective-variable probabilistic modeling, Noise handling**

I. Introduction

In its simplest form, a (supervised) classification task in data mining is to use a set of labeled data points to induce a classifier model, which can then be used to predict the label of new data points. The input data points are characterized by a number of feature values and a label, which identifies the class-value of each point. By features, we refer to the attributes or columns of a dataset and we use class to refer to the column containing the label or class-value of the data points. The classifier induced from the input data is used to find the class-value of an unlabeled data point given its feature values.

A well-known problem related to this task is to find the subset of features that should be used for determining the class-values [1]. Selecting an appropriate subset of features can reduce the overall computational complexity and improve the classification accuracy. Therefore, usually an additional step is carried out before/meanwhile learning the classifier model from data points to search for an appropriate subset of features, especially in high-dimensional problems with a large number of features.

One of the approaches to select a subset of features which has gained a lot of attention in the past few years is multiobjective optimization. An important motivation for this approach is the intrinsic conflict between problem goals (e.g. maximize accuracy and minimize model complexity) which cannot be easily aggregated to a single objective. Moreover, the space of all possible feature subsets is huge which makes it impossible to use exhaustive methods to find the optimal feature subset according to the optimization criteria. Therefore, a very good candidate for searching this space is to use stochastic heuristics. Especially, evolutionary algorithms (EAs) with their population-based search have been shown to achieve very good results in multiobjective optimization.

In a typical multiobjective optimization problem (MOP), a set of objective functions, $\mathcal{F} = \{f_1, \ldots, f_m\}$, defined over n-dimensional input vectors, should be optimized simultaneously. If we assume, without loss of generality, that all objective functions should be minimized, an MOP can be defined as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{D} \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

where $\mathcal{D}$ defines the search space of the MOP, i.e. the set of all possible solutions. The goal of an evolutionary multiobjective optimization (EMO) algorithm is to search for candidate solutions with optimal trade-off between different objective functions. These solutions are often referred to as the Pareto optimal solutions.

Usually, the objectives considered for feature subset selection (FSS) problem cannot be computed directly from the subsets. Instead, the objectives are *estimated* using a set of data points with/without a simulation process. Therefore, there is an inherent uncertainty in the objective values obtained for feature subsets, which varies depending on the method and data points used for estimation. The population-based search in EAs enables them to deal with low levels of noise in single-objective optimization problems. However, in a multiobjective scenario the noise in objective values can prevent EAs from performing an effective search in the solution space [2].

This work has been partially supported by Consolider Ingenio 2010-CSD2007-00018, TIN2010-20900-C04-04, TIN2010-14931 and Cajal Blue Brain projects (Spanish Ministry of Science and Innovation).

H. Karshenas, C. Bielza and P. Larrañaga are with the Computational Intelligence Group, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain. E-mail: {hkarshenas, mcbielza, pedro.larranaga}@fi.upm.es

Q. Zhang is with the School of Computer Science & Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK. E-mail: qzhang@essex.ac.uk

The FSS problem considered in this paper is formulated as an MOP with six objective functions. These functions which are obtained according to the performance of a classifier are the area under ROC curve, sensitivity, specificity, precision, F1 measure and Brier score. To deal with the noise in objective values, we consider an interval of values for each objective instead of a singular value. These intervals are obtained from estimating the objectives in different conditions (e.g. with different sets of data points). To select a subset of solutions, we propose a solution ranking method based on an extension of the Pareto dominance relation, which can handle objective values that are given as intervals. With this ranking method, the proposed algorithm is able to take into account the noise in objective values during optimization.

A specific type of EA, namely estimation of distribution algorithm [3], [4], is used to search in the space of possible feature subsets. In each generation the algorithm learns a joint probabilistic model from the set of selected solutions and their objective values, and use it to generate new solutions in the search space. A two-step approach is proposed for learning the joint probabilistic model. In the first step, the set of more relevant variables to the objectives are identified by learning a number of linear regression models and then combining them together. In the second step, a multidimensional Bayesian network is estimated for the objectives and the set of selected variables.

The rest of the paper is organized as follows: in Section II we first introduce the FSS problem and then review some of the EMO-based methods proposed for this problem. The details of our proposal including the description of the probabilistic model, solution ranking and joint modeling approach are explained in Section III. The formulation of the FSS problem considered in this paper is given in Section IV. Section V contains the description of the experiments, their results and related explanations. Finally, the paper is concluded in Section VI.

II. Evolutionary multiobjective algorithms in feature subset selection

A. Feature subset selection

FSS can be formally expressed as selecting the best subset of features for a learner model, given the set of all candidate features [5]. Therefore, the objective of FSS is to reduce the number of features used to characterize the dataset while improving the performance of the learner model on that dataset. According to Blum and Langley [6], an FSS method should address the following issues:

1) Initial point: the starting point(s) of the search process. It can be an empty subset (i.e. no features selected), a full subset or a randomly generated subset.
2) Search strategy: the algorithm used to explore the space of possible feature subsets. This space is exponential in the number of features (2^n), and thus FSS is considered to be a difficult combinatorial problem with an intractable computational complexity [7], [8].
3) Feature subset evaluation: measuring the quality of different feature subsets, so that high quality subsets can be preferred over others. Generally, two major approaches to feature subset evaluation exist. The *wrapper* approach uses the performance of a learner model, trained with the features in a subset, as the evaluation criteria of that subset. In the *filter* approach, instead of the learner model performance, data-driven measures (e.g. correlation, mutual information, etc.) are used to evaluate the subset of features.
4) Search stopping criteria: determine how the search method will be terminated. For example insignificant change in the quality of feature subsets, or reaching a maximum number of feature subset evaluations.

A search algorithm for FSS deals with three different spaces. First, the space of all possible feature subsets which defines the search space. This is the space that the search algorithm explores. Second, the samples in the dataset represent points in the data space. For each solution in the search space, a different projection of the data space, obtained according to the features included in that solution, is used to evaluate the solution (e.g. by training and testing a learner model). Third, the result of evaluating each solution is a point in the objective space. Viewed in this way, feature subset evaluation is a function that maps each solution in the search space through data space to a point in the objective space.

Both wrapper and filter approaches have been used for evaluating feature subsets in multiobjective FSS with EAs. To evaluate a solution (i.e. feature subset) within a wrapper approach, first a learner model is trained from a training dataset only using the features included in that solution. Then this model is tested on a separate validation or test dataset to assess its performance. To increase the accuracy of the assessment, usually this process is repeated several times. Techniques like bootstrapping, k-fold cross-validation or leave-one-out (a special version of the latter) are used for partitioning the dataset and repeated evaluation of a learner model.

Although this way of evaluating solutions has a high computational complexity, but from the learner model performance perspective, often the solutions found with this approach are superior to those found with filter-based methods. Therefore, most of the EMO algorithms for FSS are based on a wrapper approach, using k-fold cross-validation of a usually simple learner model with small training time. Some of the methods have also used distributed evaluation to speed up solution evaluation [9] or approximation techniques to prevent retraining the classifier for each single solution [10], [11].

Techniques like bootstrapping and k-fold cross-validation can obtain a good estimation of the quality of each solution. However, more accurate estimation of quality can be obtained by testing the final solutions found by the search algorithm on an independent dataset which is not used during the search algorithm. On the other hand, to have a statistical estimation of a search algorithm performance in FSS, the algorithm should be run several times. Thus, to combine these two requirements for quality and performance assessment, several bi-partitions of the given dataset are considered. For each bi-partition, two individual runs of the search algorithm are performed. In the first run, one of the partitions is used to evaluate the solutions

during the search process (e.g. by k-fold cross-validation) and the other partition is used to test the final solutions after the search. In the second run the role of the partitions is exchanged.

In the following two sections we briefly review some of the works in the literature that use EMO algorithms for FSS. These works are summarized in TABLE I. The intuitive type for representing a feature subset, adopted by all of the methods reviewed here, is to use a binary encoding, i.e. a bit string of length equal to the number of features, where a zero value means the exclusion of the corresponding feature from the subset and a value of one means its inclusion. Thus, there is a one to one correspondence between the features of the dataset and the variables in the solutions to the FSS problem.

B. FSS in classification

Emmanouilidis et al. [12] proposed a commonality-based crossover in the niched Pareto genetic algorithm (NPGA) [13], where common alleles of the parent solutions are directly copied to the offspring solutions, and the other genes are inherited based on a probability computed from the number of common genes. They used the classification accuracy of artificial neural networks (ANN) along with the size of feature subset as the objectives of optimization. Also, they applied their method to the rotating machinery fault diagnosis problem with respectively two (ANN approximated root mean squared error and feature subset size) [10] and three objectives (sensitivity and specificity of a nearest neighbor (NN) classifier along with the size of feature subset) [14].

Oliveira et al. [11] used the first version of non-dominated sorting GA based on fitness sharing (NSGA) [15] to select a good subset of features for handwritten digit recognition problem. The classification accuracy of an ANN classifier and the size of feature subsets were used as optimization objectives. They also used EMO to search for the best ensemble of the classifiers found in the Pareto set after FSS search [9].

Some other works on the use of EMO for FSS are based on the second version of non-dominated sorting GA (NSGA-II) [16]. Shi et al. [17] tried to find the best feature subsets for an ensemble of support vector machines (SVMs) in the protein fold recognition problem. Three objectives were used for optimization: 1) cross-validation classification accuracy, 2) test classification accuracy, and 3) feature subset size. Hamdani et al. [18] studied the performance of NSGA-II for FSS on several datasets of varying sizes. They used the classification accuracy of an NN classifier and feature subset size as optimization objectives. Ekbal et al. [19] searched for relevant features in named entity recognition problem of natural language processing with a wrapper method using maximum entropy-based classifiers. Recall and precision of the classifiers were used as objectives during the search process, and F-measure was used to select one of the feature subsets from the final Pareto set. Huang et al. [20] performed separate optimizations for each of the feature subset sizes in the problem of predicting customer churn in telecommunications. Classification accuracy, true positive rate and true negative rate of a decision tree (DT) classifier were used as objectives in each of the optimization runs.

In a different context, Rodríguez and Lozano [21] used NSGA-II to perform a multiobjective search for the best structure of a multidimensional Bayesian classifier which involves selecting the subset of features relevant to each class variable. They used the classification accuracy of each of the classes as the optimization objectives. Radtke et al. [22] proposed a three phase multiobjective optimization for: 1) feature extraction, 2) single classifier or ensemble components selection, and 3) FSS for improving the performance of the selected classifier or ensemble. They compared the performance of NSGA-II and a multiobjective memetic algorithm (that uses local search) on the handwritten digit recognition problem with respect to classification accuracy and feature subset size as objectives.

Zhu et al. [23] used a hybrid wrapper-filter approach by combining wrapper-based NSGA-II and filter-based local search in a memetic algorithm. The accuracies of each of the class-values in a one-versus-all classification scheme, obtained from DT classifiers, were used as optimization objectives in NSGA-II, whereas the criterion for local search was based on the feature-class relevance. Spolaôr et al. [24] proposed several filter-based bi-objective optimizations with NSGA-II for FSS, each time pairing interclass distance measure with one of the following criteria: ratio of inconsistent pairs of samples in the dataset, feature-class correlation, Laplacian score of the samples in the dataset and features entropy.

Very recently, Vatolkin et al. [25] employed a hypervolume indicator-based EMO algorithm to search for good feature subsets in the high-dimensional problem of musical instruments recognition in a polyphonic audio mixture. They used the relative feature subset size and classification mean squared error as optimization objectives. DTs, random forests, naïve Bayes and SVMs were used as alternative classifiers to compute the second objective.

C. FSS in clustering

In unsupervised learning or clustering context, the solution encoding may be extended to also include the number of clusters. Thus, the algorithm will be simultaneously searching the space of possible feature subsets and the space of possible cluster numbers. The learner model in this context is a clustering algorithm which assigns the samples in the dataset to a number of clusters. The objectives here usually are to increase the closeness of the samples in the same cluster (cluster compactness), while increasing the separation between different clusters.

Kim et al. [26] proposed an evolutionary local search algorithm (ELSA) to select the proper subset of features for clustering. They proposed four objectives when using the K-means algorithm and three objectives when using the expectation maximization (EM) algorithm. Morita et al. [27] used NSGA to cluster handwritten month names with the K-means algorithm and two objectives. Handl and Knowles [28] proposed a general framework for bi-objective FSS in clustering problems which encompasses both filter and wrapper approaches. Their framework is based on the second version of Pareto envelope-based selection algorithm (PESA-II) [29] and uses the K-means algorithm for clustering in the wrapper approach.

TABLE I
SUMMARY OF THE METHODS FOR FSS USING EMO ALGORITHMS.

		Approach	Optimizer	Learner Model	# Objectives
Classification	Emmanouilidis et al., 2000 [12]	Wrapper	NPGA	ANN	2
	Emmanouilidis et al., 2001 [10]	Wrapper	NPGA	ANN	2
	Emmanouilidis, 2001 [14]	Wrapper	NPGA	1-NN	3
	Oliveira et al., 2002 [11]	Wrapper	NSGA	ANN	2
	Oliveira et al., 2006 [9]	Wrapper	NSGA	ANN	2
	Shi et al., 2004 [17]	Wrapper	NSGA-II	Ensemble of SVMs	3
	Hamdani et al., 2007 [18]	Wrapper	NSGA-II	1-NN	2
	Rodríguez and Lozano, 2008 [21]	Wrapper	NSGA-II	Multidimensional Bayesian classifier	2
	Ekbal et al., 2010 [19]	Wrapper	NSGA-II	Max entropy-based classifier	2
	Huang et al., 2010 [20]	Wrapper	NSGA-II	DT	3
	Radtke et al., 2009 [22]	Wrapper	NSGA-II, Memetic EMO	ANN, Projection distance-based classifier	2
	Vatolkin et al., 2012 [25]	Wrapper	Indicator-based EMO	DT, Random forest, Naïve Bayes, SVM	2
	Zhu et al., 2009 [23]	Hybrid	Memetic EMO	DT	3, 4
	Spolaôr et al., 2011 [24]	Filter	NSGA-II	—	2
Clustering	Kim et al., 2002 [26]	Wrapper	ELSA	K-means, EM	3, 4
	Morita et al., 2003 [27]	Wrapper	NSGA	K-means	2
	Zhang et al., 2006 [30]	Wrapper	Immunology-based EMO	Fuzzy c-means	3
	Handl and Knowles, 2006 [28]	Wrapper, Filter	PESA-II	K-means	2
	Zaharie et al., 2007 [31]	Filter	NSGA-II	—	4

Instead of representing the feature subsets with bit strings, Zhang et al. [30] encode feature saliencies in real-valued strings and select only those features with a saliency value above a given threshold. They employed an EA inspired by immunology to solve a three objective optimization problem by using a fuzzy c-means algorithm for clustering. Zaharie et al. [31] used NSGA-II in a filter approach to find the best ranking of the features, a closely related problem to FSS. They considered four different objectives for optimization.

III. MULTIOBJECTIVE FSS USING JOINT MODELING OF OBJECTIVES AND VARIABLES

A. Multiobjective estimation of distribution algorithms

One of the recent paradigms in evolutionary computation is estimation of distribution algorithms (EDAs), developed to overcome the shortcomings in traditional EAs. In each generation, an EDA learns a probabilistic model from the set of selected solutions, acquiring an abstraction of the common properties of those solutions. This probabilistic model is then used to generate new solutions in the search space. Many variants of EDAs, using different types of probabilistic models, have been proposed and they are successfully applied to a variety of problem domains. For a review of some of these methods, the interested reader is referred to [32], [33].

EDAs have also been used for multiobjective optimization and several multiobjective EDAs are proposed in the literature [34]–[37]. These methods usually integrate solution ranking and selection mechanisms of EMO algorithms into the framework of EDAs based on estimating and sampling a probabilistic model to perform the search in the multiobjective space.

In this paper, we employ a specific multiobjective EDA for FSS which is based on joint modeling of objectives and variables, called the multidimensional Bayesian network-based EDA (MBN-EDA) [38]. The probabilistic model used in this algorithm is a type of Bayesian network [39] that allows to learn complex patterns of interaction between constituting variables. In our approach to multiobjective optimization, this probabilistic model can encode not only the relationships between variables, but also those between objectives and between objectives and variables (an implicit variable selection for each objective). This extra information allows MBN-EDA to take into account the estimated qualities of solutions when generating new solutions. The analysis of the probabilistic models learnt during different generations, presented later on, shows that the algorithm considers these new types of relationships more important for multiobjective search than the relationships between variables.

In the following sections we explain how MBN-EDA has been adapted for FSS. Our proposal comprises two major parts corresponding to the main steps of a typical EDA, each explained in a different section. Fig. 1 shows the overall outline of the proposed algorithm for FSS.

B. Solution ranking

Often, the well-known Pareto dominance relation [40] is used to order the solutions in multi-objective optimization. Mathematically, this relation states that a solution $\boldsymbol{x}$ dominates another solution $\boldsymbol{y}$, denoted as $\boldsymbol{x} \prec \boldsymbol{y}$, if and only if $f_j(\boldsymbol{x}) \leq f_j(\boldsymbol{y})$, $\forall f_j \in \mathcal{F}$, and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ for at least one $f_j \in \mathcal{F}$. However, a common property of real-world problems is the existence of uncertainty, found in the measurements, modeling or evaluation of real systems. One of the important consequences of uncertainty in such problems is that the objective values returned for each solution involve noise. Optimization in this kind of noisy environments can mislead the search and eventually prevent the optimization algorithm from obtaining a good estimation of the Pareto optimal set. For example, in EMO algorithms based on Pareto dominance relation, the set of non-dominated solutions maintained in

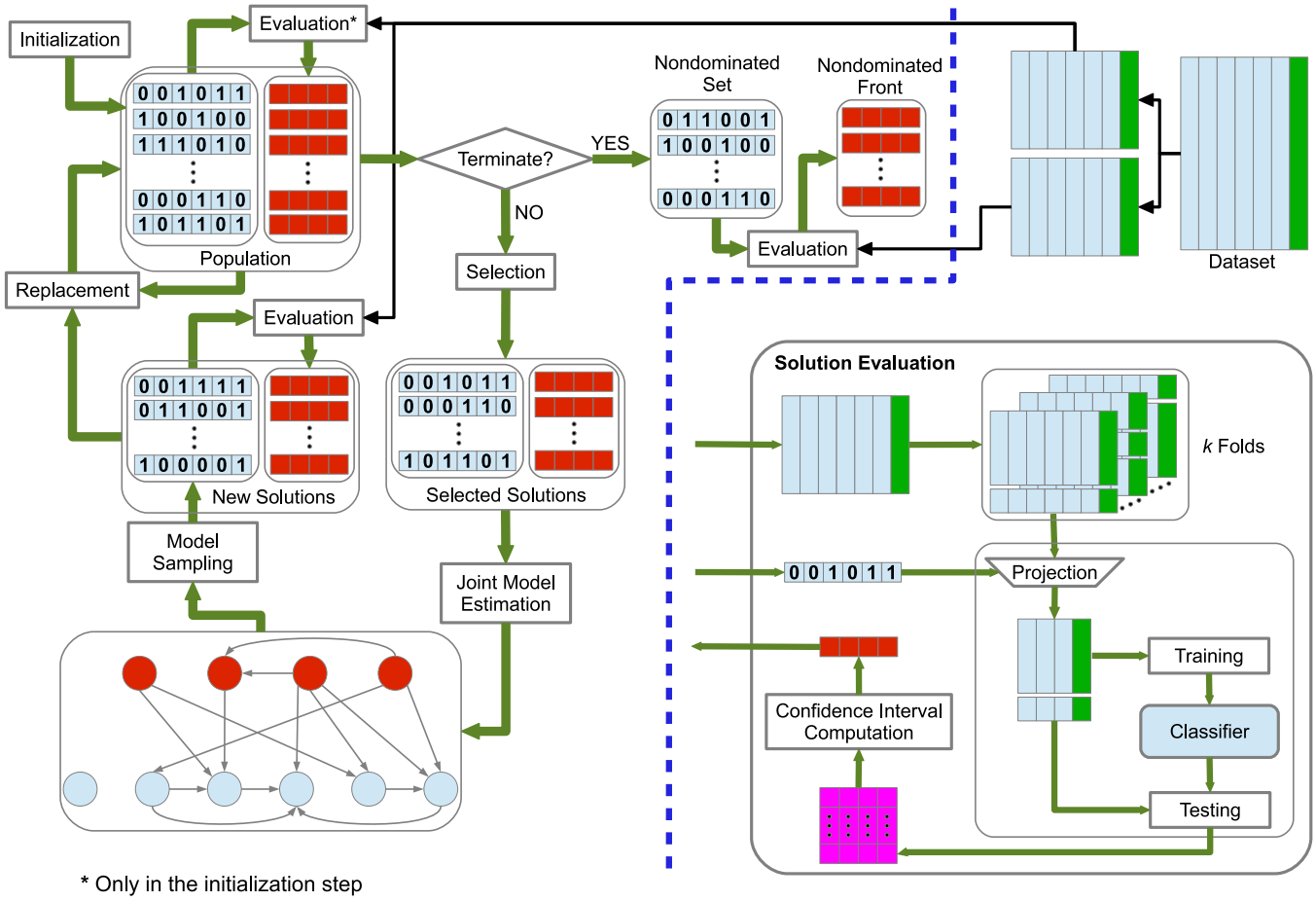


Fig. 1. The overview of the proposed EDA for FSS in classification.

each generation may contain some solutions that are actually dominated, or the algorithm can discard solutions that are non-dominated if the true objective values were taken into account. This inaccuracy in finding non-dominated sets can eventually affect algorithm convergence to Pareto optimal front.

Dealing with noisy objective values in EMO has been extensively studied in the recent years (e.g. see [2] for a review). The basic idea in these methods is to modify solution evaluation and ranking so that the effect of noise can be taken into account. One of the approaches, adopted by many of the methods, is to reevaluate each solution several times and obtain a statistical estimation of the objective values, for example by taking the mean value of each objective. In some problem formulations, however, solution evaluation inherently involves multiple reevaluations. For example, when evaluating feature subsets using k-fold cross-validation or bootstrapping, a set of values are estimated for each of the objectives.

Instead of estimating a singular value from these reevaluations as the value of each objective, an alternative approach, which has been less studied so far, is to assume that each objective returns a set or interval of values. In this way the algorithm can also take into account the noise in the objective values when selecting a subset of solutions. Probabilistic dominance [41], [42], is a well-known noise handling method in EMO which extends the traditional Pareto dominance relation to objectives with interval values. It assumes that the set of objectives returned for each solution, $\boldsymbol{F}(\boldsymbol{x})$, is a vector of random variables and computes the probability that the objective vector of a solution $\boldsymbol{x}$ dominates the objective vector of another solution $\boldsymbol{y}$, i.e. $P\big(\boldsymbol{F}(\boldsymbol{x}) \prec \boldsymbol{F}(\boldsymbol{y})\big)$.

Based on this dominance probability, Hughes [42] proposed a probabilistic ranking (PR) method of the solutions in the population:

$$rank_{PR}(\boldsymbol{x}_i) = \sum_{k=1}^{N} P\big(\boldsymbol{F}(\boldsymbol{x}_k) \prec \boldsymbol{F}(\boldsymbol{x}_i)\big) + \frac{1}{2}\sum_{k=1}^{N} P\big(\boldsymbol{F}(\boldsymbol{x}_k) \equiv \boldsymbol{F}(\boldsymbol{x}_i)\big) - \frac{1}{2}, \quad (2)$$

where

$$P\big(\boldsymbol{F}(\boldsymbol{x}) \equiv \boldsymbol{F}(\boldsymbol{y})\big) = 1 - P\big(\boldsymbol{F}(\boldsymbol{x}) \prec \boldsymbol{F}(\boldsymbol{y})\big) - P\big(\boldsymbol{F}(\boldsymbol{x}) \succ \boldsymbol{F}(\boldsymbol{y})\big) \quad (3)$$

represents the probability when neither of the objective vectors of the solutions can dominate each other. The last term in Equation (2) is subtracted so that the sum of the ranks of all the solutions in the population equals to $\frac{N(N-1)}{2}$, where N is the number of solutions in the population. This ranking method defines a total order between solutions.

One of the drawbacks of probabilistic dominance relation is the computational complexity of calculating the probabilities. Therefore, it is often assumed that the random variables of the objective vectors are statistically independent, and thus this probability can be computed as:

$$P\big(\boldsymbol{F}(\boldsymbol{x}) \prec \boldsymbol{F}(\boldsymbol{y})\big) = \prod_{j=1}^{m} P\big(f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})\big). \quad (4)$$

In this paper, we introduce a ranking method based on a different dominance relation. This dominance relation is an extension of the traditional Pareto dominance relation to cases where the values of some objectives are given as an interval. Thus, we partition the set of objective functions $\mathcal{F}$ into two disjoint subsets of objective functions with singular values $\mathcal{F}_S$ and noisy objective functions with interval values $\mathcal{F}_I$, i.e. $\mathcal{F}_S \cup \mathcal{F}_I = \mathcal{F}$ and $\mathcal{F}_S \cap \mathcal{F}_I = \emptyset$.

Definition 1. **(α-Degree Pareto Dominance)** *Let $L\big(f_j(\boldsymbol{x})\big)$ and $U\big(f_j(\boldsymbol{x})\big)$ represent, respectively, the lower and upper bounds of the interval returned for solution $\boldsymbol{x}$ by the noisy objective function $f_j \in \mathcal{F}_I$. Then, solution $\boldsymbol{x}$ is said to dominate another solution $\boldsymbol{y}$ with a degree $\alpha \in (0, 1]$, denoted as $\boldsymbol{x} \prec_\alpha \boldsymbol{y}$, if and only if:*

1) $\forall f_j \in \mathcal{F}_S \quad f_j(\boldsymbol{x}) \leq f_j(\boldsymbol{y})$, *and*
2) $\forall f_j \in \mathcal{F}_I \quad deg_j(\boldsymbol{x}, \boldsymbol{y}) \geq \alpha$, *and*
3) $\big(\exists f_k \in \mathcal{F}_S \quad f_k(\boldsymbol{x}) < f_k(\boldsymbol{y}) \quad \vee$
$\exists f_k \in \mathcal{F}_I \quad deg_k(\boldsymbol{x}, \boldsymbol{y}) > \alpha\big)$,

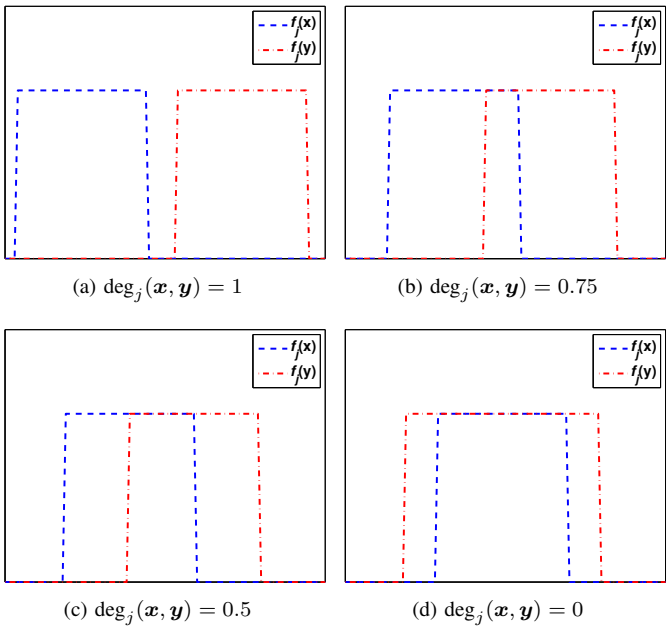
where $deg_j(\boldsymbol{x}, \boldsymbol{y})$ is the degree that solution $\boldsymbol{x}$ dominates solution $\boldsymbol{y}$ with respect to the noisy objective $f_j \in \mathcal{F}_I$:

$$deg_j(\boldsymbol{x}, \boldsymbol{y}) = \min\Big\{1, \max\big\{0, \frac{L\big(f_j(\boldsymbol{y})\big) - L\big(f_j(\boldsymbol{x})\big)}{U\big(f_j(\boldsymbol{x})\big) - L\big(f_j(\boldsymbol{x})\big)}\big\}\Big\}. \quad (5)$$

Intuitively, $deg_j(\boldsymbol{x}, \boldsymbol{y})$ computes the percentage of the interval obtained for solution $\boldsymbol{x}$ that is not overlapped by the interval obtained for solution $\boldsymbol{y}$ in objective $f_j \in \mathcal{F}_I$, confined within $[0, 1]$. Thus, only that segment of the solution $\boldsymbol{x}$ interval which is better than the best point in the solution $\boldsymbol{y}$ interval (its lower bound in minimization) is taken into account.

Definition 1 allows a solution to dominate other solutions when its corresponding intervals are partially better (according to the $deg_j(\cdot, \cdot)$ function) than the intervals of other solutions. Fig. 2 shows some examples of two intervals placement and the corresponding values of the $deg_j(\cdot, \cdot)$ function. With higher values of α, a solution can only dominate other solutions if major segments of its intervals are better than the intervals corresponding to other solutions, thus placing a stricter condition for accepting a solution as non-dominated.

Similar to traditional Pareto dominance relation, it can be proven that α-degree Pareto dominance relation defines a partial order between the solutions. Therefore, terms like α-degree Pareto optimal solution, α-degree Pareto non-dominated set, α-degree Pareto optimal set and α-degree Pareto optimal front can be similarly defined and adopted for MOPs with noisy objectives which take on interval values. Moreover, well-known solution ranking methods proposed in the literature like non-dominated sorting algorithm [16] can be straightforwardly used for MOPs with noisy objectives.



(a) $\deg_j(\boldsymbol{x}, \boldsymbol{y}) = 1$ (b) $\deg_j(\boldsymbol{x}, \boldsymbol{y}) = 0.75$

(c) $\deg_j(\boldsymbol{x}, \boldsymbol{y}) = 0.5$ (d) $\deg_j(\boldsymbol{x}, \boldsymbol{y}) = 0$

Fig. 2. Examples of interval values and the resulting degree of dominance.

Here, we propose a degree ranking (DR) method of the solutions in the population based on α-degree Pareto dominance. In this method, first the solutions are sorted into a number of non-dominated sets using α-degree Pareto dominance and then their ranks are computed as

$$rank_{DR}(\boldsymbol{x}_i) = \theta_{ND} \cdot r + \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{m} deg_j(\boldsymbol{x}_k, \boldsymbol{x}_i) + \theta_B \cdot \sum_{j=1}^{m} I\big(f_j(\boldsymbol{x}_i), f_j^*\big), \quad (6)$$

where r is the rank of the non-dominated set containing solution $\boldsymbol{x}_i$ (starting from 1 for the best non-dominated set), $I(\mathcal{B}, a)$ is an indicator function returning one if the best point in interval $\mathcal{B}$ is equal to a and zero otherwise, and f_j^* is the best value reached so far in objective f_j. θ_{ND} and θ_B are the coefficients that determine the importance of respectively non-domination ranks and best-found boundary values in solution ranking.

This ranking method combines the measures of solution convergence and diversity with non-dominated ranks. The computation of these measures are inspired by gain-based and distance to best-based ranking methods proposed in the literature [43], [44]. The second and third terms in Equation (6) can be computed while sorting the solutions into non-dominated sets (with a complexity of order $O(N^2 m)$) and thus do not impose additional computational overhead like for example in the computation of crowding distances. Since solution ranking is one of the most time consuming steps in EMO algorithms, this reduction in computational time is highly favorable.

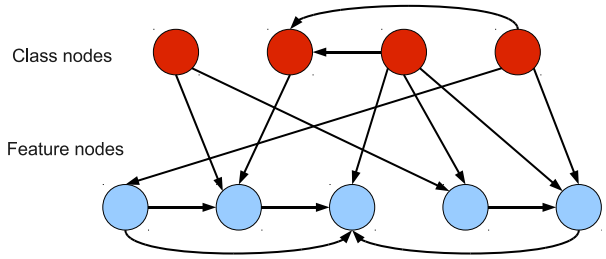


Fig. 3. An example of an MBN structure, used for multi-dimensional classification

C. Probabilistic modeling

The probabilistic model used for the joint modeling of objectives and variables is a multidimensional Bayesian network (MBN) [45]. This type of Bayesian network is usually used in multi-dimensional classification where a sample of dataset can simultaneously belong to several classes. Fig. 3 shows an example of an MBN structure. The nodes in the structure are organized in two separate layers: the top layer comprises class variables and the bottom layer contains feature variables. The set of arcs is partitioned into three subsets, resulting in the following subgraphs:

- the class subgraph, containing the class nodes and the interactions between them,
- the feature subgraph, comprising the feature variables and their relations, and
- the bridge subgraph, depicting the *one-way* dependencies from class nodes to feature nodes.

To represent an MOP with an MBN in our approach, the variables are modeled as feature nodes and the objectives as class nodes. The feature subgraph encodes the relationships between problem variables and the class subgraph represents the relationships between objectives. The bridge subgraph shows which variables are related to each objective, implicitly performing a kind of FSS for each objective. We use this property to initialize the MBN learning algorithm in our approach.

As explained in Section II, FSS is a combinatorial problem with discrete search domain where candidate solutions are presented with bit strings. However, the objectives considered for this problem are usually continuous-valued functions or, as assumed in Section III-B, interval-valued functions. Therefore, learning a joint model of objectives and variables can become overly complex and computationally very demanding. Since the main purpose of joint modeling in our algorithm is to obtain an approximation of the interactions between objectives and variables, we use estimations of objective values to simplify model learning.

The method we propose for learning a joint model of objectives and variables involves two major parts. Fig. 4 shows the outline of our approach. In the first part, a linear regression model is estimated for each objective function to find the most related subset of variables to that objective. Formally, let $\big(\boldsymbol{x}_i, \boldsymbol{F}(\boldsymbol{x}_i)\big) = \big(\boldsymbol{x}_i, f_1(\boldsymbol{x}_i), \cdots, f_m(\boldsymbol{x}_i)\big)$ denote a solution of the population and its corresponding objective values. Then, we learn an ℓ_1 regularized regression model (RRM) [46], [47] for each objective f_j given the variables, minimizing the following penalized sum of squared errors:

$$\sum_{i=1}^{N} \Big(\hat{\mathbb{E}}\big(f_j(\boldsymbol{x}_i)\big) - \big(\beta_{j0} + \boldsymbol{\beta}_j \boldsymbol{x}_i^{\mathrm{T}}\big)\Big)^2 + \lambda \sum_{k=1}^{n} |\beta_{jk}|. \quad (7)$$

In this equation, the approximated expected values of each objective are used to estimate the RRM parameters, which are the intercept β_{j0} and the regression coefficients $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jn})$. The last summand in Equation (7) is called the regularization term and parameter $\lambda > 0$ determines the regularization intensity. This type of regularization, which is also called least absolute shrinkage and selection operator (LASSO), has the promising property of setting some of the regression coefficients exactly to zero, thus excluding the corresponding variable from RRM. As a result, a subset of variables are selected for each of the objectives.

The RRM learning algorithm finds solutions to Equation (7) for a range of λ values, considering different regularization intensities. For each objective, we select the model which has the highest score according to the Akaike information criterion (AIC) [48], a typical model selection metric in regression.

Next, the final subset of variables selected for each of the objectives are combined to obtain a common subset of most relevant features to all objectives. Strategies like taking the union or the intersection of the variable subsets are proposed in the literature [49]. Here, we adopt an intermediate approach by selecting those variables that have appeared in at least half the RRMs. This divides the set of variables into two groups: those selected in the combined model, $\boldsymbol{X}_S$, and the rest, $\bar{\boldsymbol{X}}_S = \boldsymbol{X} \setminus \boldsymbol{X}_S$.

In the second part, a greedy local search algorithm is used to learn an MBN of the objectives and variables in $\boldsymbol{X}_S$. For this purpose, first the objective values are discretized, using

---

Inputs:
Selected solutions $\boldsymbol{D}_{\boldsymbol{X}}$
Their objective values $\boldsymbol{Q}_{\mathcal{F}}$

// First part
$\boldsymbol{Q}^{\mathbb{E}}_{\mathcal{F}} \leftarrow$ Estimate expected values of objectives from $\boldsymbol{Q}_{\mathcal{F}}$
for all $f_j \in \mathcal{F}$ **do**
  $\mathcal{M}_R[\boldsymbol{X}, f_j] \leftarrow$ Estimate an RRM from $\big(\boldsymbol{D}_{\boldsymbol{X}}, \boldsymbol{Q}^{\mathbb{E}}_{f_j}\big)$
end for
$\boldsymbol{X}_S \leftarrow$ Combine variables selected in $\mathcal{M}_R[\boldsymbol{X}, f_j], \forall f_j \in \mathcal{F}$
$\bar{\boldsymbol{X}}_S \leftarrow \boldsymbol{X} \setminus \boldsymbol{X}_S$
$\mathcal{S}_I[\boldsymbol{X}_S, \mathcal{F}] \leftarrow$ Initialize to empty structure
for all $f_j \in \mathcal{F}$ **do**
  $\mathcal{M}_R[\boldsymbol{X}_S, f_j] \leftarrow$ Remove variables in $\bar{\boldsymbol{X}}_S$ from $\mathcal{M}_R[\boldsymbol{X}, f_j]$
  $\mathcal{S}_I[\boldsymbol{X}_S, \mathcal{F}] \leftarrow \mathcal{S}_I[\boldsymbol{X}_S, \mathcal{F}]+$ Structure of $\mathcal{M}_R[\boldsymbol{X}_S, f_j]$
end for

// Second part
$\boldsymbol{Q}^{D}_{\mathcal{F}} \leftarrow$ Discretize objective values in $\boldsymbol{Q}^{\mathbb{E}}_{\mathcal{F}}$
$\mathcal{M}_1[\boldsymbol{X}_S, \mathcal{F}] \leftarrow$ Estimate an MBN from $\big(\boldsymbol{D}_{\boldsymbol{X}_S}, \boldsymbol{Q}^{D}_{\mathcal{F}}\big)$ starting from structure $\mathcal{S}_I[\boldsymbol{X}_S, \mathcal{F}]$
for all $X_i \in \bar{\boldsymbol{X}}_S$ **do**
  $\mathcal{M}_2[X_i] \leftarrow$ Estimate univariate probability distribution from $\boldsymbol{D}_{X_i}$
end for

Output: $\big(\mathcal{M}_1[\boldsymbol{X}_S, \mathcal{F}], \{\mathcal{M}_2[X_i] \mid X_i \in \bar{\boldsymbol{X}}_S\}\big)$

Fig. 4. Outline of the joint model estimation method.

an equal frequency discretization method, into three nominal values: good, average and bad. Then, the structure of the combined model, obtained in the previous part, is used to initialize the bridge subgraph of MBN before starting the search. In each iteration, the search algorithm checks all valid arc addition, removal and reversal operations, and applies the operation with the highest increase in the MBN score [50]. The Bayesian information criterion (BIC) [51] is used to score MBN structures, which accounts for the likelihood of the data but penalizes the number of model parameters, thus preferring simpler models.

The variables in $\bar{\boldsymbol{X}}_S$, which are considered less important to the objectives, can be ignored in the modeling process, thus copying their values when generating new solutions. However, to allow the possibility of future participation in joint modeling for these variables (in the next generations), we estimate a univariate probability distribution for each variable in $\bar{\boldsymbol{X}}_S$ [52] to explore the subspace of these variables with low-complexity modeling.

At the end, the joint probability distribution of variables and objectives, encoded in the estimated MBN and individual univariate probability distributions, is given by

$$P\big(x_1,\ldots,x_n,q_1,\ldots,q_m\big) = \prod_{X_i\in\boldsymbol{X}_S} P(x_i|\boldsymbol{pa}(X_i)) \cdot \prod_{j=1}^{m} P(q_j|\boldsymbol{pa}(f_j)) \cdot \prod_{X_k\in\bar{\boldsymbol{X}}_S} P(x_k), \tag{8}$$

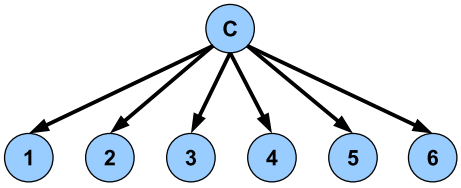
where $\boldsymbol{Pa}(X_i) \subseteq \{\mathcal{F}\cup\boldsymbol{X}_S\setminus X_i\}$ and $\boldsymbol{Pa}(f_j)\subseteq\{\mathcal{F}\setminus f_j\}$ respectively are the parents of each variable and objective in the MBN structure, and $\boldsymbol{pa}(X_i)$ and $\boldsymbol{pa}(f_j)$ represent one of their possible instantiations. q_j denotes a (discrete) instantiation of objective f_j. The first and second terms in Equation (8) correspond to the probability distribution encoded in the MBN and the third term is the joint probability distribution of the variables in $\bar{\boldsymbol{X}}_S$.

Finally, to generate new candidate solutions, all the variables ($\boldsymbol{X}$) are sampled from the joint probability distribution in Equation (8) by taking the objective values of the selected solutions as evidences. Further details of sampling the estimated MBN can be found in [38].

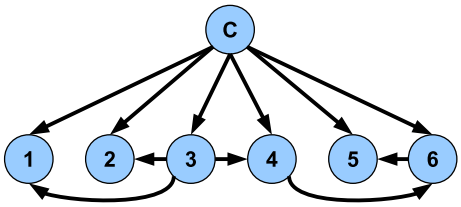
In the joint modeling approach proposed here, we have used the variable selection property of ℓ_1 regularization to remove those variables that are less important to the set of objective functions being optimized. This reduction in the number of variables in the first part, reduces the computational complexity of learning the joint model of objectives and variables in the second part. Moreover, the structure estimated from RRMs in the first part for the relationships between variables and objectives, is an approximation of the MBN's bridge subgraph and therefore can serve as a good initial searching point for the (reduced) space of possible MBN structures.

IV. Problem Formulation

This section describes how FSS is formulated in this paper. We adopt a wrapper approach to evaluate feature subsets using two different Bayesian classifiers. It should be noted that these two classifiers define two different optimization problems on each dataset since the best feature subset obtained for one type of classifier might not necessarily be the best feature subset for another type of classifier.



(a) NB Classifier



(b) TAN Classifier

Fig. 5. Sample structure of the classifiers used for solution evaluation.

The first classifier is naïve Bayes (NB) [53] which assumes that features are statistically independent given the class. Fig. 5a shows an example of NB classifier structure. The probability of a specific class-value c given feature values $\boldsymbol{x}$ is given by

$$P(c\mid\boldsymbol{x}) \propto P(c)P(x_1,\ldots,x_n\mid c) = P(c)P(x_1\mid c)\cdots P(x_n\mid c). \tag{9}$$

The classifier training only consists of finding the prior probability of class-values and the conditional univariate probabilities of each feature values. Despite its simple structure, NB classifier is shown to have very good classification performance in many real-world problems.

The second classifier is tree-augmented naïve Bayes (TAN) [54] which represents the relationships between features with a tree structure. An example of TAN classifier structure is depicted in Fig. 5b. This classifier's training algorithm involves finding the maximum weighted spanning tree over the features, based on their conditional mutual information given the class. It is proven that this algorithm learns the maximum log-likelihood TAN classifier for a given dataset [54].

These two classifiers choose the class-value with the highest posterior probability as the label of a given instance $\boldsymbol{x}$ of features. We use six different performance measures for the classifiers based on the classification accuracy, given by a confusion matrix and class-value probabilities. These measures are: sensitivity, specificity, precision, area under receiver operating characteristics curve (AUC), F1 and Brier score, and are computed as follows:

$$f_{AUC} = \frac{G_1+1}{2},$$
$$f_{sens} = \frac{TP}{TP+FN},$$
$$f_{spec} = \frac{TN}{TN+FP},$$

TABLE II
DATASETS CONSIDERED FOR THE EXPERIMENTS

Name	# samples	# features	# class-values	Missing values
WDBC	569	30	2	No
Ozone	2536	72	2	Yes
Hill-Valley	2424	100	2	No

$$
\begin{aligned}
f_{prec} &= \frac{TP}{TP+FP},\\
f_{F1} &= \frac{2TP}{2TP+FN+FP},\\
f_{Brier} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}\big(p(c_k \mid \boldsymbol{x}_i) - \delta(c_k,\boldsymbol{x}_i)\big)^2,
\end{aligned} \tag{10}
$$

where G_1 is the Gini coefficient estimated using the area of a number of trapezoids. TP, TN, FP and FN are the entries of the confusion matrix showing respectively the number of true positive, true negative, false positive and false negative samples classified by the model. N is the total number of samples, i.e. $N = TP + TN + FP + FN$. C is the number of possible class-values: $\{c_1, \ldots, c_C\}$, and function $\delta(c_k, \boldsymbol{x}_i)$ returns one if the true class-value of instance $\boldsymbol{x}_i$ is c_k and zero otherwise.

These measures define the objective functions of optimization. The first five objectives should be maximized and have a range of values in $[0, 1]$ whereas the last objective should be minimized and represents the calibration error in classification.

Following the common practice in FSS literature, we use binary encoding to represent feature subsets with bit strings of length equal to the number all features. Each feature subset is evaluated using k-fold cross-validation of a classifier on the given dataset, projected over the features in the subset. Thus, we will obtain k values for each of the classifier performance measures (objectives). We then use these values to compute a confidence interval with a confidence level $\gamma \in [0, 1]$ for each of the performance measures.

V. EXPERIMENTS

A. Experimental design

We have used three datasets with increasing number of features, to study the performance of our proposal for FSS. These datasets, which are all retrieved from UCI online machine learning repository[1], are Wisconsin diagnostic breast cancer (WDBC), ozone level detection (Ozone) and Hill-Valley, with details presented in Table II. To handle the missing values in the Ozone dataset, we discard samples with missing class-value, or if half the features are missing. Otherwise, the missing values are replaced with the mean value of that feature.

To evaluate the performance of the algorithm, 5 random bi-partitions of each dataset are generated (elsewhere this method is called 5×2 cross-validation), resulting in 10 independent runs. For each run, the number of cross-validation folds is set to $k = 5$ and a confidence level of $\gamma = 0.95$ is used to compute the intervals for the performance measures (objectives). The evaluation of the final non-dominated set of each run on the independent test set is used as the final non-dominated front obtained in that run.

Three quality indicators are used to inspect the convergence, diversity and distribution of the final non-dominated fronts obtained by the algorithm. These indicators are respectively hypervolume [55], maximum spread [56] and Schott's spacing [57], computed using the expected value of the objectives. Hypervolume indicator computes the volume covered by the non-dominated front with respect to a reference point. Larger values of this indicator show better approximations. Maximum spread indicator gives an estimation of the non-dominated front diversity by taking into account the minimum and maximum values achieved for each objective. Larger values of this indicator show a more diverse front and are desired. Schott's spacing indicator is a measure of how the points are distributed over the non-dominated front. It is based on the distance between each point and its closest neighbor in the objective space. Lower values of this metric are favored.

The initial population of MBN-EDA is generated randomly with a uniform distribution. The full set of features and the empty subset are also added to the initial population. The truncation selection mechanism with a threshold of $\tau = 0.5$ is used to select a subset of solutions for new offspring generation. An elitist replacement mechanism is adopted to add the newly generated solutions to the population, which selects the best solutions from offspring and population solutions. The population size is set to $N = 2,000$ to allow better estimations of MBN parameters. In the MBN estimation process, the maximum number of parents for each node is set to $\max\{m, \lceil \log_3(\tau N) \rceil\}$, allowing the possibility for the variable nodes to have all objectives as their parents. In our experiments, the MBN learning algorithm virtually never needed to surpass this maximum (in less than 0.15% of model estimations an operation was canceled because of reaching this maximum). To compensate for the large population size requirement of MBN-EDA, we set the maximum number of generations to 50.

B. MBN-EDA with different solution ranking methods

In the following experiments we use our DR method to rank the solutions for selection and compare it with the previously proposed PR method. To simplify PR computation of solutions, we use an approximation of the dominance probability in each objective assuming a Gaussian noise [42]:

$$
\begin{aligned}
P\big(f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})\big) &= \frac{1}{2} - \frac{1}{2}\mathrm{erf}\Big(\frac{f_j(\boldsymbol{x}) - f_j(\boldsymbol{y})}{\sigma_b\sqrt{2+2(\frac{\sigma_a}{\sigma_b})^2}}\Big)\\
&\approx \frac{1}{2} - \frac{1}{2}\tanh\Big(\frac{f_j(\boldsymbol{x}) - f_j(\boldsymbol{y})}{\sigma_b\sqrt{2+2(\frac{\sigma_a}{\sigma_b})^2}}\Big),
\end{aligned} \tag{11}
$$

where σ_a and σ_b are the standard deviations of the values of $f_j(\boldsymbol{x})$ and $f_j(\boldsymbol{y})$, respectively.

Two points should be taken into consideration when choosing the values for the combination coefficients of the DR method in Equation (6): 1) solutions in lower-ranked fronts should generally receive lower ranks than the solutions in

[1] http://archive.ics.uci.edu/ml/datasets.html

higher-ranked fronts, and 2) solutions close to the best value found for any of the objectives should be preferred to advocate fronts with larger diversity. After testing different values for these coefficients, they are set to $\theta_{ND} = 2$ and $\theta_B = 2$ in the experiments. We study three different dominance degrees (α) for the DR method: 0.1, 0.5 and 0.9.

Fig. 6–8 show results of the quality indicators obtained for the final non-dominated fronts with these ranking methods, on the three datasets considered in this study. For the WDBC dataset (Fig. 6), the PR method has a slightly better average performance with respect to all three indicators and for both classifiers. Comparing the results obtained with different α values in the DR method we see that, according to hypervolume indicator, higher α values result in better non-dominated fronts. Higher α values place stricter requirements for a solution to dominate the others, resulting in fewer non-dominated solutions but with higher degree of reliability. Thus, in the presence of noise in the objective values, this method can lead to better convergence of the non-dominated fronts.

For Ozone and Hill-Valley datasets with large search spaces, where the level of noise in the objective values can increase, the non-dominated fronts obtained with the DR method are better spread than with the PR method, resulting in higher hypervolume values especially for the NB classifier. It can be seen in Fig. 8 that lower diversity of the fronts obtained for Hill-Valley dataset by the PR method causes small spacing between the solutions in these fronts. This means that with the PR method the search is focused on a smaller region of the space. In general, smaller spacing is favorable in the comparison of two non-dominated fronts if they have similar diversity.

Fig. 9 shows the time required by each of the ranking methods to order the solutions for replacement. The times are averaged over the generations of each run and over the three datasets. It can be seen that DR method based on α-degree Pareto dominance requires significantly less time (less than half) than the PR method based on probabilistic dominance, even when using the approximation in Equation (11). This time is not directly dependent on the specific dataset or classifier used for evaluation. Rather, the choice of dataset and classifier influences the objective values of the solutions, creating different instances of the population to be ranked.

Although NB and TAN classifiers define two different optimization problems for each dataset, the quality indicators show that the feature subsets found for NB classifier result in better overall classification performance. It seems that with TAN classifier the level of noise in objective values is higher. A closer look at the feature subsets in the final non-dominated sets also show that fewer features are selected for the NB classifier (Fig. 10). Moreover, for this classifier usually a similar number of features are selected in the final non-dominated sets found by MBN-EDA with both PR and DR methods (the latter with different α values). For the TAN classifier, on the other hand, larger feature subsets are selected that are manly different with each of the ranking methods, especially for the Hill-Valley dataset.

The range of the objective values in the aggregation of the final non-dominated fronts, obtained with different ranking methods are given in TABLE III. These aggregated fronts are obtained by taking the non-dominated solutions from the union of the non-dominated sets found in 10 different runs. For the WDBC dataset, the range of objective values are small and very close to their optimal values. The sensitivity and F1 measure of the solutions in the final non-dominated sets

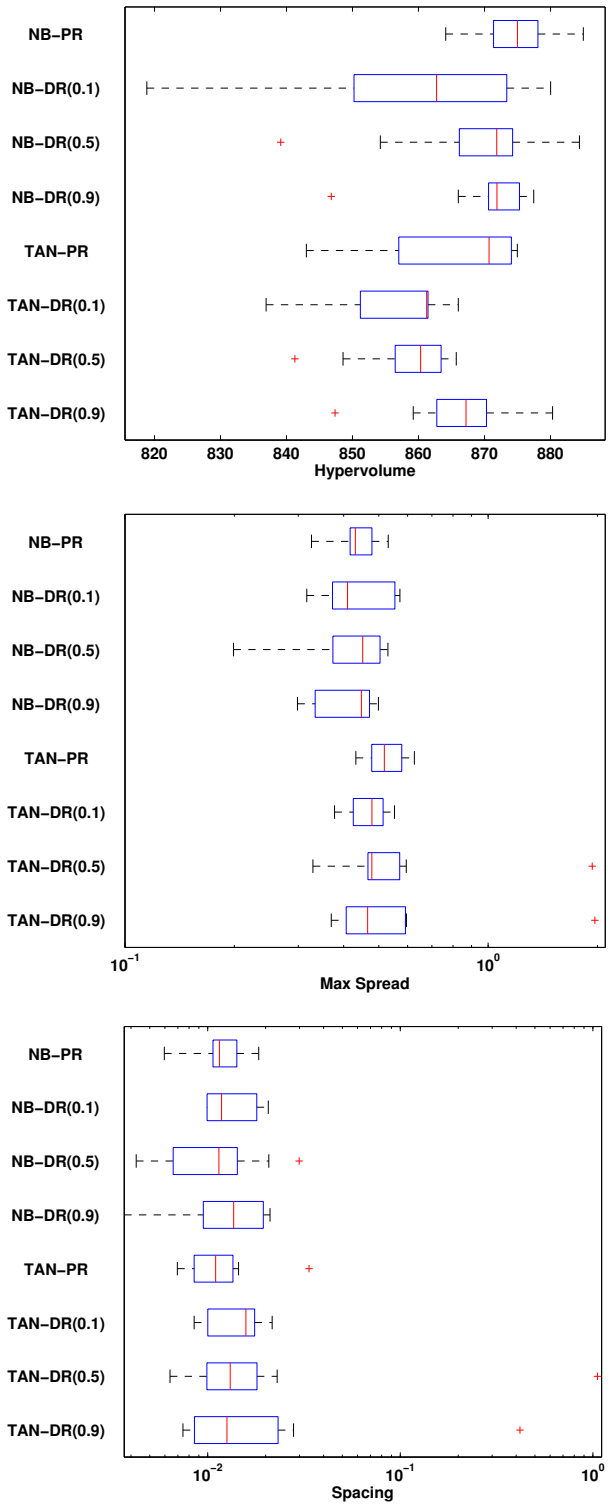
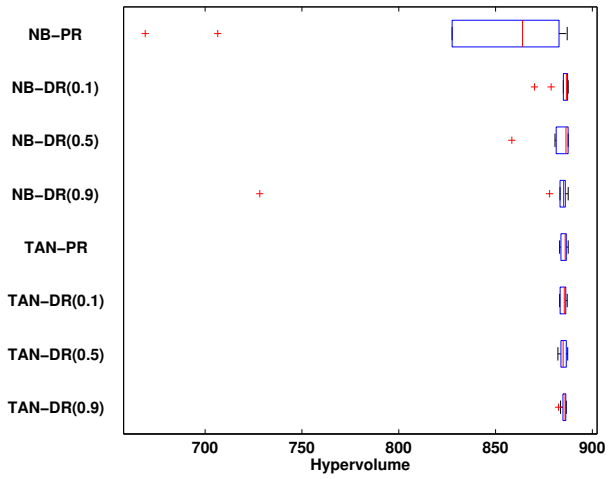
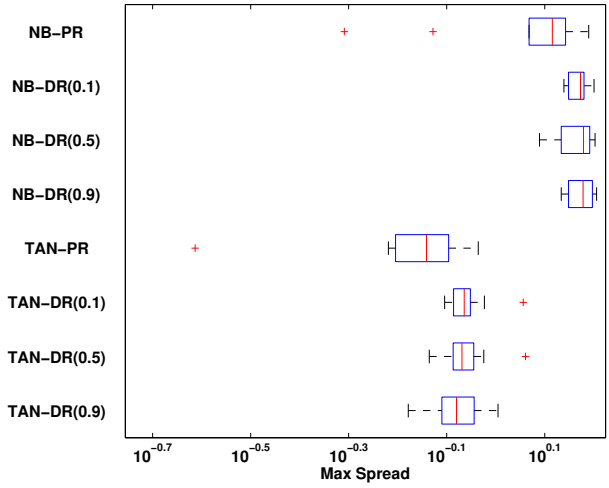


Fig. 6. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for NB and TAN classifiers on WDBC dataset using PR and DR ranking methods in MBN-EDA.





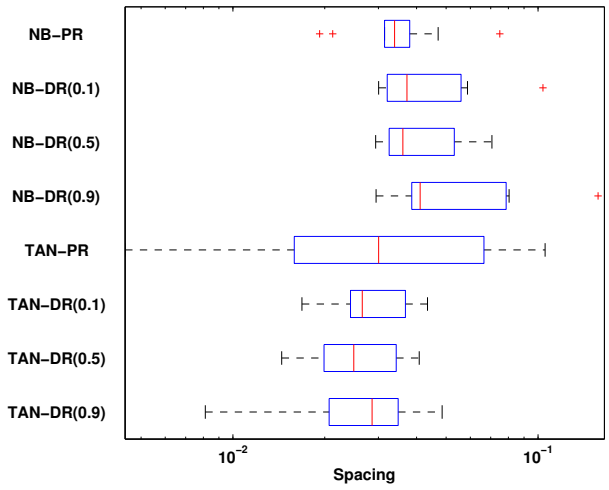
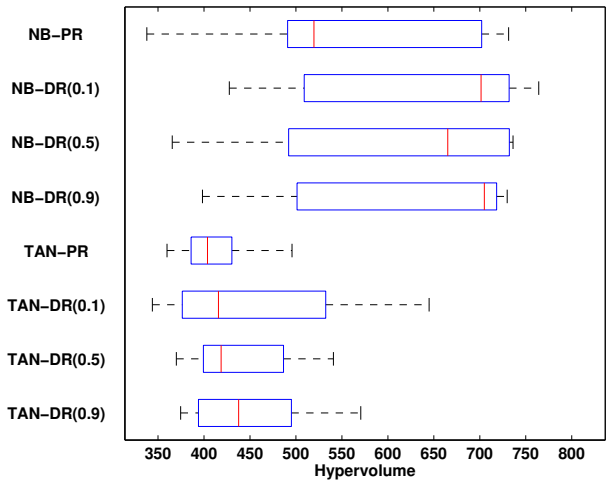
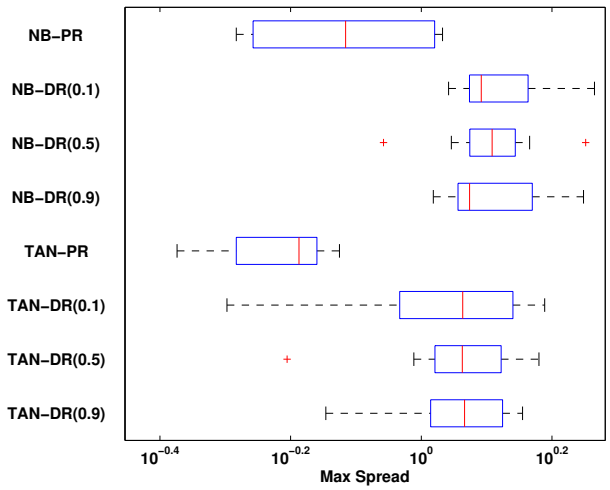


Fig. 7. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for NB and TAN classifiers on Ozone dataset using PR and DR ranking methods in MBN-EDA.





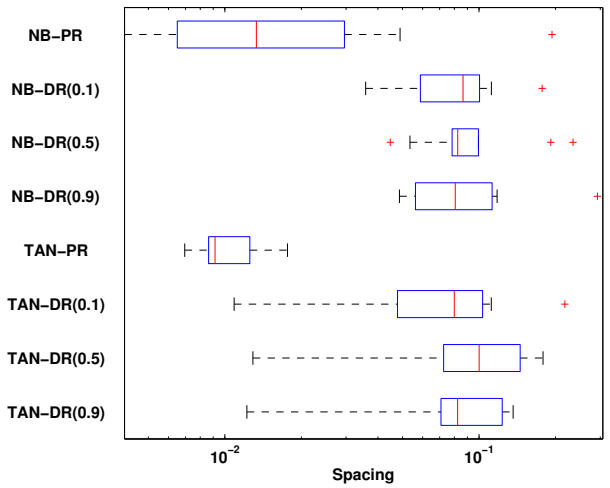


Fig. 8. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for NB and TAN classifiers on Hill-Valley dataset using PR and DR ranking methods in MBN-EDA.

seem to be slightly worse than other classifier performance measures.

As the noise in objective values increases for the Ozone and Hill-Valley datasets, making these problems more difficult, the range of objective values of the non-dominated fronts also increases and gets farther from the optimal values. First, for the Ozone dataset, the highly unbalanced samples (less than 3% of the samples are positive) affect the sensitivity, precision and F1 measure functions which are based on the number of TP samples. This influence can be specially observed when using TAN classifier to evaluate solutions, where despite very small sensitivity, they have small Brier scores, explaining the good

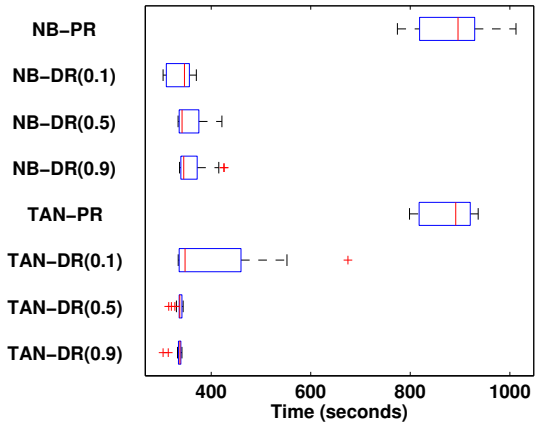


Fig. 9. Average solution ranking time in each generation of MBN-EDA with PR and DR ranking methods.

results of hypervolume indicator for this dataset and classifier.

Second, the sensitivity and specificity of the solutions found for the Hill-Valley dataset, cover a large range of possible values for these objectives, whereas their AUC and Brier score indicate poor performances. This shows that for larger search spaces, sensitivity and specificity functions take priority over the other objectives in the optimization process. As it can be seen, considering several performance measures for evaluating the feature subsets allows us to inspect the final solutions from different points of view, which would not be possible when using only one or two objectives.

C. Comparison with genetic algorithm

To evaluate the proposed model estimation in MBN-EDA, we have compared it with a standard genetic algorithm (GA) and studied their optimization performance. Since the recombination operators of GA do not require large populations as in MBN-EDA, we have set the population size of GA to $N = 300$ and allowed the algorithm to evolve for more generations by setting the maximum number of generations to 350. Thus, while GA and MBN-EDA are using two different strategies for evolution, both of them have access to similar resources when considering the maximum number of function evaluations. The GA considered in the experiments employs a two-point crossover and bit-flip mutation with probabilities $P_{cross} = 0.8$ and $P_{mut} = 1/n$, respectively. The rest of parameters like the selection ratio are set similar to MBN-EDA as described in the previous section.

Fig. 11–13 compare the final non-dominated fronts obtained by GA and MBN-EDA on each of the three datasets with different quality indicators, when using the NB classifier. Very similar results are also obtained for the TAN classifier. The figures show that the fronts found by MBN-EDA are better than or comparable to the fronts obtained by GA on all datasets with respect to different quality indicators. Especially, for the WDBC and Ozone datasets, the hypervolume of the fronts obtained by MBN-EDA is considerably better. This indicates that, although MBN-EDA evolves fewer generations, it is able to perform a more effective search using its joint probabilistic model.

Also, these figures show that the non-dominated fronts found by GA using the DR method result in better hypervolume values than those found using the PR method, when considering the overall behavior on all datasets. This suggests that, in spite of the method used to explore the search space, the solution ranking provided by the DR method can often help

TABLE III
THE RANGE OF OBJECTIVE VALUES OF THE SOLUTIONS IN THE TOTAL NON-DOMINATED SET, OBTAINED FROM AGGREGATING THE FINAL NON-DOMINATED SETS OF 10 INDEPENDENT MBN-EDA RUNS, FOR THE THREE DATASETS.

WDBC		f_{AUC}	f_{sens}	f_{spec}	f_{prec}	f_{F1}	f_{Brier}
NB-PR	Min.	0.974	0.897	0.958	0.917	0.912	0.038
	Max.	0.998	0.970	0.994	0.990	0.961	0.111
NB-DR(0.1)	Min.	0.978	0.864	0.948	0.920	0.903	0.068
	Max.	0.991	0.937	0.989	0.980	0.944	0.097
NB-DR(0.5)	Min.	0.980	0.911	0.968	0.936	0.923	0.054
	Max.	0.987	0.955	0.984	0.967	0.954	0.084
NB-DR(0.9)	Min.	0.979	0.920	0.957	0.924	0.925	0.059
	Max.	0.991	0.941	0.989	0.982	0.954	0.090
TAN-PR	Min.	0.958	0.838	0.912	0.851	0.856	0.077
	Max.	0.994	0.958	0.978	0.957	0.940	0.166
TAN-DR(0.1)	Min.	0.966	0.875	0.918	0.856	0.863	0.085
	Max.	0.988	0.944	0.961	0.935	0.936	0.154
TAN-DR(0.5)	Min.	0.969	0.874	0.903	0.848	0.861	0.081
	Max.	0.988	0.937	0.972	0.952	0.940	0.179
TAN-DR(0.9)	Min.	0.976	0.880	0.924	0.891	0.896	0.030
	Max.	0.996	0.979	0.990	0.981	0.979	0.132

Ozone		f_{AUC}	f_{sens}	f_{spec}	f_{prec}	f_{F1}	f_{Brier}
NB-PR	Min.	0.763	0.000	0.807	0.000	0.000	0.054
	Max.	0.902	0.696	1.000	0.700	0.350	0.341
NB-DR(0.1)	Min.	0.720	0.000	0.818	0.000	0.000	0.048
	Max.	0.919	0.783	1.000	0.433	0.360	0.309
NB-DR(0.5)	Min.	0.793	0.000	0.798	0.000	0.000	0.052
	Max.	0.916	0.674	1.000	0.527	0.314	0.364
NB-DR(0.9)	Min.	0.777	0.000	0.824	0.000	0.000	0.054
	Max.	0.922	0.748	1.000	0.662	0.423	0.309
TAN-PR	Min.	0.686	0.000	0.992	0.000	0.000	0.051
	Max.	0.861	0.092	1.000	0.467	0.124	0.073
TAN-DR(0.1)	Min.	0.721	0.000	0.990	0.000	0.000	0.052
	Max.	0.845	0.164	1.000	0.467	0.208	0.076
TAN-DR(0.5)	Min.	0.757	0.000	0.989	0.000	0.000	0.054
	Max.	0.842	0.130	1.000	0.600	0.157	0.070
TAN-DR(0.9)	Min.	0.690	0.000	0.991	0.000	0.000	0.052
	Max.	0.874	0.096	1.000	0.467	0.140	0.077

Hill-Valley		f_{AUC}	f_{sens}	f_{spec}	f_{prec}	f_{F1}	f_{Brier}
NB-PR	Min.	0.465	0.073	0.228	0.317	0.118	0.526
	Max.	0.546	0.741	0.903	0.591	0.548	0.904
NB-DR(0.1)	Min.	0.461	0.000	0.057	0.000	0.000	0.500
	Max.	0.543	0.924	1.000	0.594	0.641	0.933
NB-DR(0.5)	Min.	0.449	0.000	0.082	0.000	0.000	0.500
	Max.	0.541	0.899	1.000	0.585	0.621	0.932
NB-DR(0.9)	Min.	0.438	0.000	0.000	0.000	0.000	0.500
	Max.	0.546	1.000	1.000	0.578	0.675	0.945
TAN-PR	Min.	0.488	0.343	0.346	0.457	0.387	0.513
	Max.	0.591	0.723	0.665	0.598	0.617	0.605
TAN-DR(0.1)	Min.	0.444	0.000	0.000	0.000	0.000	0.501
	Max.	0.581	1.000	1.000	0.573	0.690	0.682
TAN-DR(0.5)	Min.	0.480	0.000	0.000	0.000	0.000	0.500
	Max.	0.582	1.000	1.000	0.579	0.685	0.690
TAN-DR(0.9)	Min.	0.462	0.000	0.000	0.000	0.000	0.499
	Max.	0.575	1.000	1.000	0.576	0.685	0.697

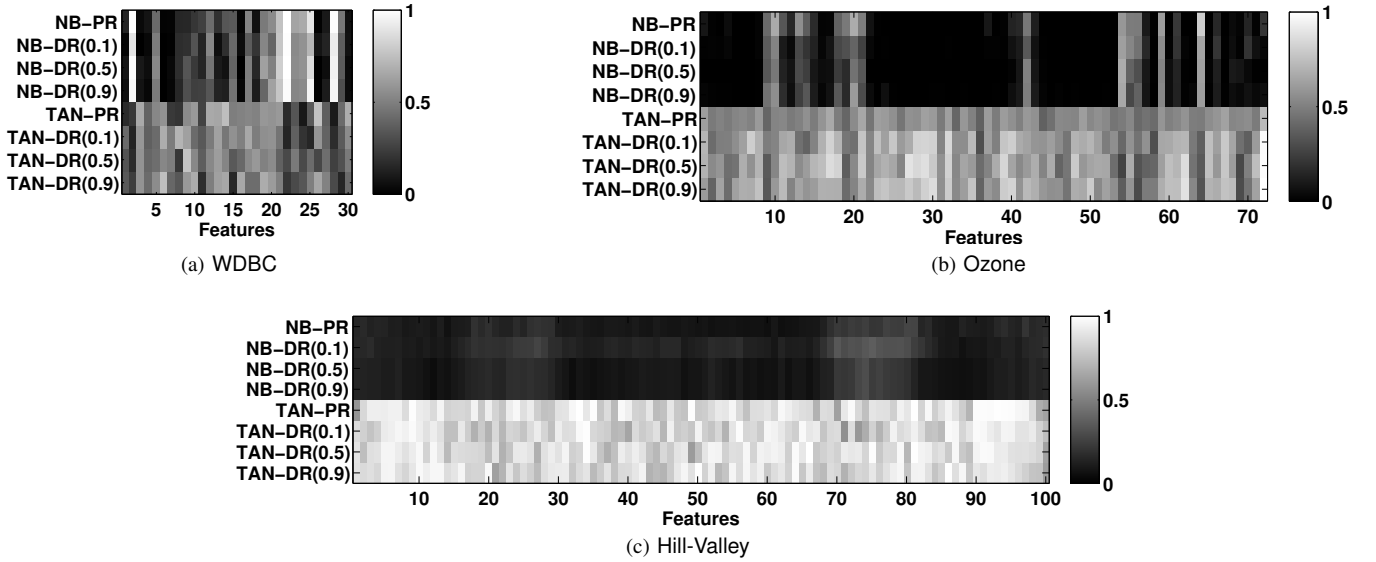


Fig. 10. The rate of selecting each feature in the solutions of the final non-dominated fronts obtained for NB and TAN classifiers on three datasets using different ranking methods in MBN-EDA. The rates are averaged over 10 independent runs.

to converge to better non-dominated fronts in noisy problems.

D. Analysis of joint probabilistic modeling

One of the advantages of EDAs is that apart from finding a solution to the optimization, they estimate a probabilistic model which captures certain regularities of the solutions and problem search space. This kind of meta information is especially useful when the intrinsic properties of the problem at hand are not very well known. Specifically, in multiobjective optimization, the estimated model encodes common properties of the approximated Pareto set and, in decision making, can be used together or even instead of the non-dominated solutions, e.g. when the size of approximated Pareto set is very large.

Recently, several works have studied the use of data mining techniques to obtain new knowledge from the set of Pareto non-dominated solutions after optimization [58], [59]. However, multiobjective EDAs can already obtain this kind of information during optimization, depending on the probabilistic model they use. In addition to variables, the joint probabilistic modeling proposed in this paper encompasses the objectives of the approximated Pareto front. This type of model can give an approximation of the problem structure (relationships between objectives and input variables) and the interactions between objective functions.

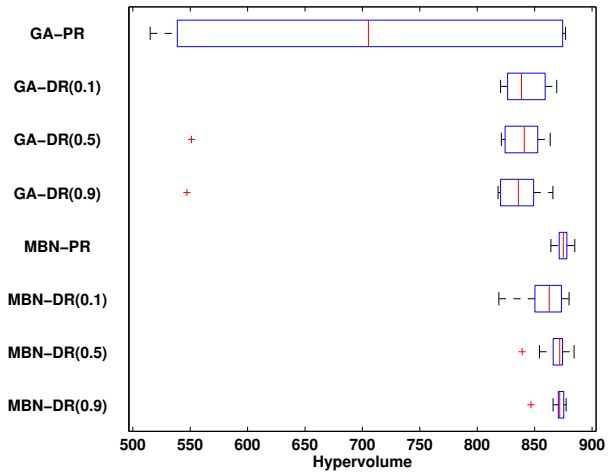
In this section, we study two constituent parts of the probabilistic models estimated in MBN-EDA during evolution. For this purpose, we consider the models estimated using the DR method with a dominance degree of $\alpha = 0.9$. First, the set of variables selected in the first part of joint model estimation using RRMs are examined. Fig. 14 shows the selected variables during different phases of evolution for the three datasets and two classifiers. It can be seen that for NB classifier, initially most of the variables are selected and during evolution, gradually the set of the chosen variables becomes smaller (except for Hill-Valley dataset which does not exactly follow this pattern). On the contrary, for TAN classifier the variable selection frequency usually increases over time.

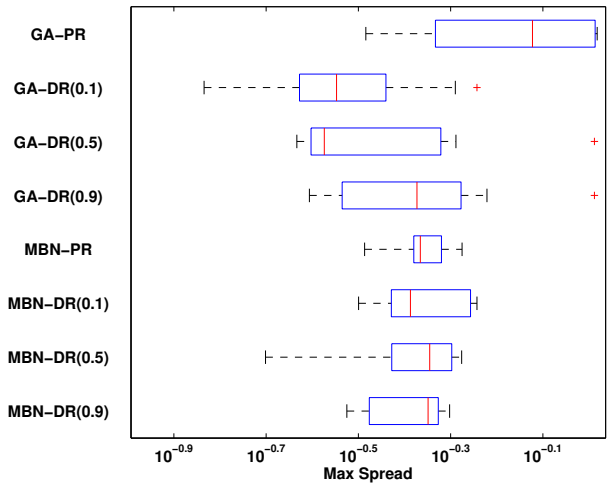
Here, the selection frequency of variables determines their relevance to the computation of objective values. Unlike the feature selection frequency obtained from the solutions, this relevance is not affected only by a certain value (for example a zero value) of the variables. In other words, both inclusion and exclusion of a specific feature are involved in approximating its relevance to objectives.

Secondly, we study the MBN structures estimated in the second part of the joint modeling algorithm. As expected, model estimation adds considerably more arcs to class and bridge subgraphs of MBN (i.e. between objectives and between objectives and variables) than to the feature subgraph, indicating the importance of these relations. Instead of depicting the complete MBN structure which requires a lot of space, we have concentrated on the class subgraphs of the estimated MBNs. Fig. 15 shows the frequency of arcs in the class subgraph of the MBN structures estimated over all generations of MBN-EDA in 10 independent runs, for the three datasets and two classifiers.

Certain patterns of interaction between objectives have a higher frequency of occurrence in different datasets. These include the dependencies between sensitivity, specificity and precision, between AUC and Brier score, and between sensitivity and F1 function. Some of these relationships can be easily approved by looking at the definitions of objective functions in Equation (10). For example, both sensitivity and F1 functions depend on the number of TP and FN samples, and thus any information on the value of one of these objectives can be used to approximate the value of the other.

An interesting observation is the role of classifiers in detecting the interactions between objectives. It seems that the proposed joint model estimation is able to identify these kind of relationships better when the TAN classifier is used for solution evaluation. One explanation for this behavior is that for the TAN classifier the sets of variables selected in the first part of model learning using RRMs are smaller, especially at the early generations of the evolution, where the algorithm





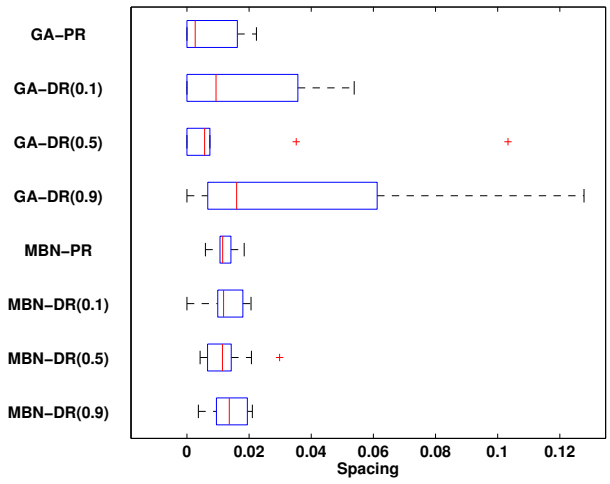
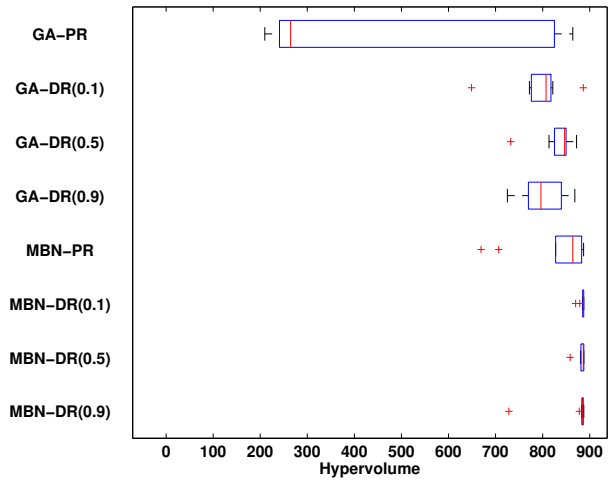
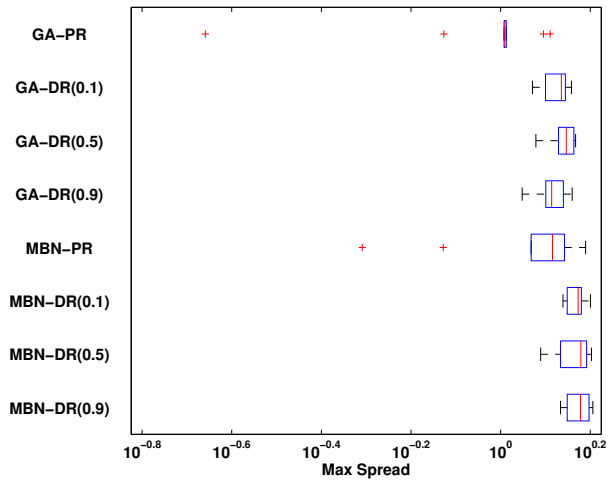


Fig. 11. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for WDBC dataset with NB classifier, using GA and MBN-EDA (denoted as MBN).





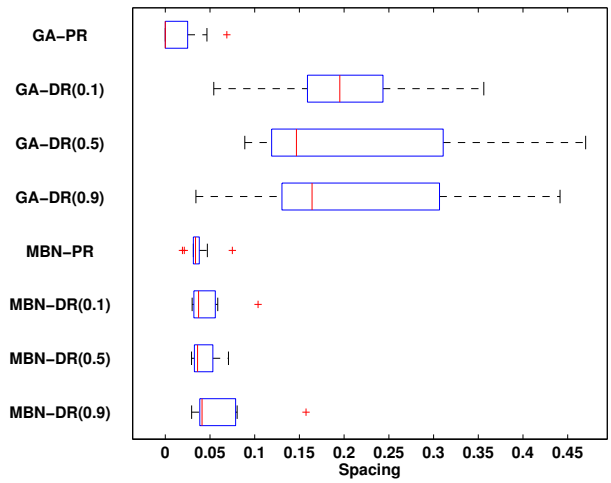


Fig. 12. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for Ozone dataset with NB classifier, using GA and MBN-EDA (denoted as MBN).

is detecting the direction of movement in the search space. This allows to filter out variables that would introduce noise to the MBN induction process, which in turn helps to detect the relationships between objectives.

For some of the problem instances under study, the probability of recovering the relationships between objectives in the joint model estimation is very low. This situation can be observed for Ozone dataset with TAN classifier (Fig. 15b, right) and Hill-Valley dataset with NB classifier Fig. 15c, left. These cases show two possible situations where objective relationships are not considered to be important for optimization by MBN-EDA. In the latter case, most of the

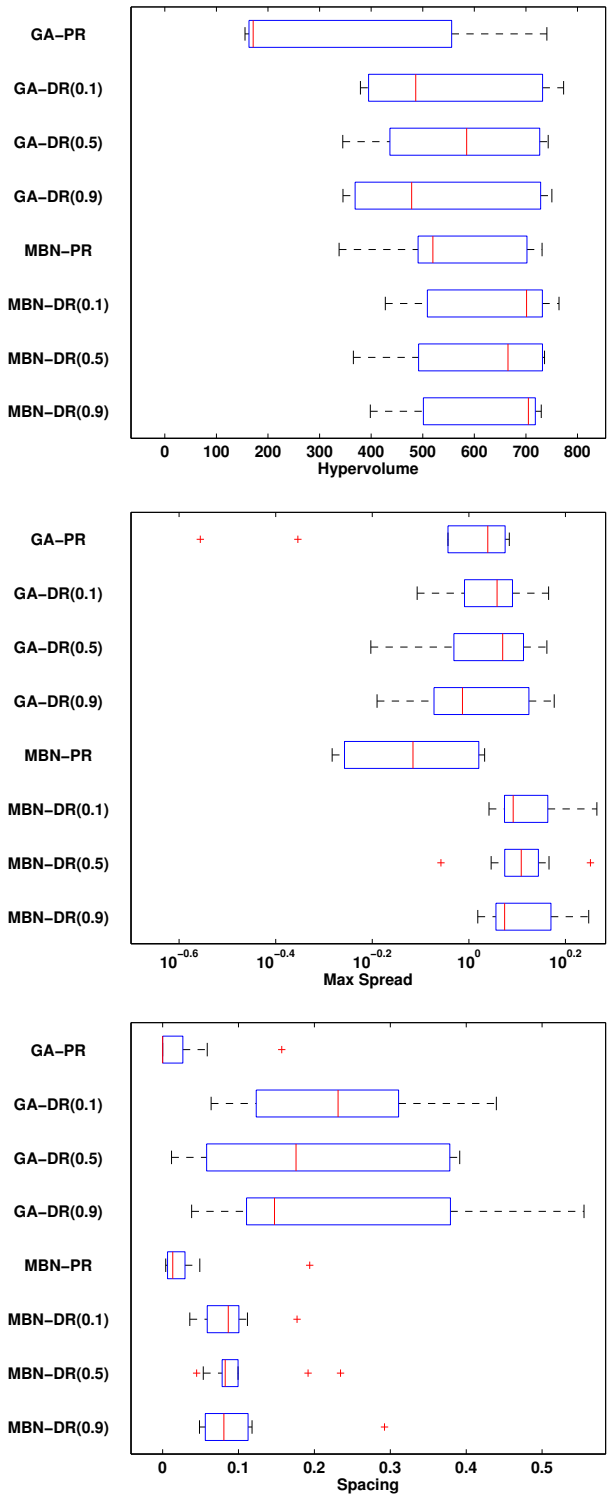


Fig. 13. Hypervolume, maximum spread and spacing of the final non-dominated fronts obtained for Hill-Valley dataset with NB classifier, using GA and MBN-EDA (denoted as MBN).

variables are selected for inclusion in MBN by the first part model estimation algorithm, in all generations (see Fig. 14). As it was already explained, this directly affects the detection of relationships in the class subgraph. In the former case, although a relatively smaller number of variables are selected for inclusion in the MBN, but the high level of inconsistency in the objective values due to noise (especially with the unbalanced dataset) makes it hard for the MBN induction algorithm to detect these interactions.

VI. Conclusions

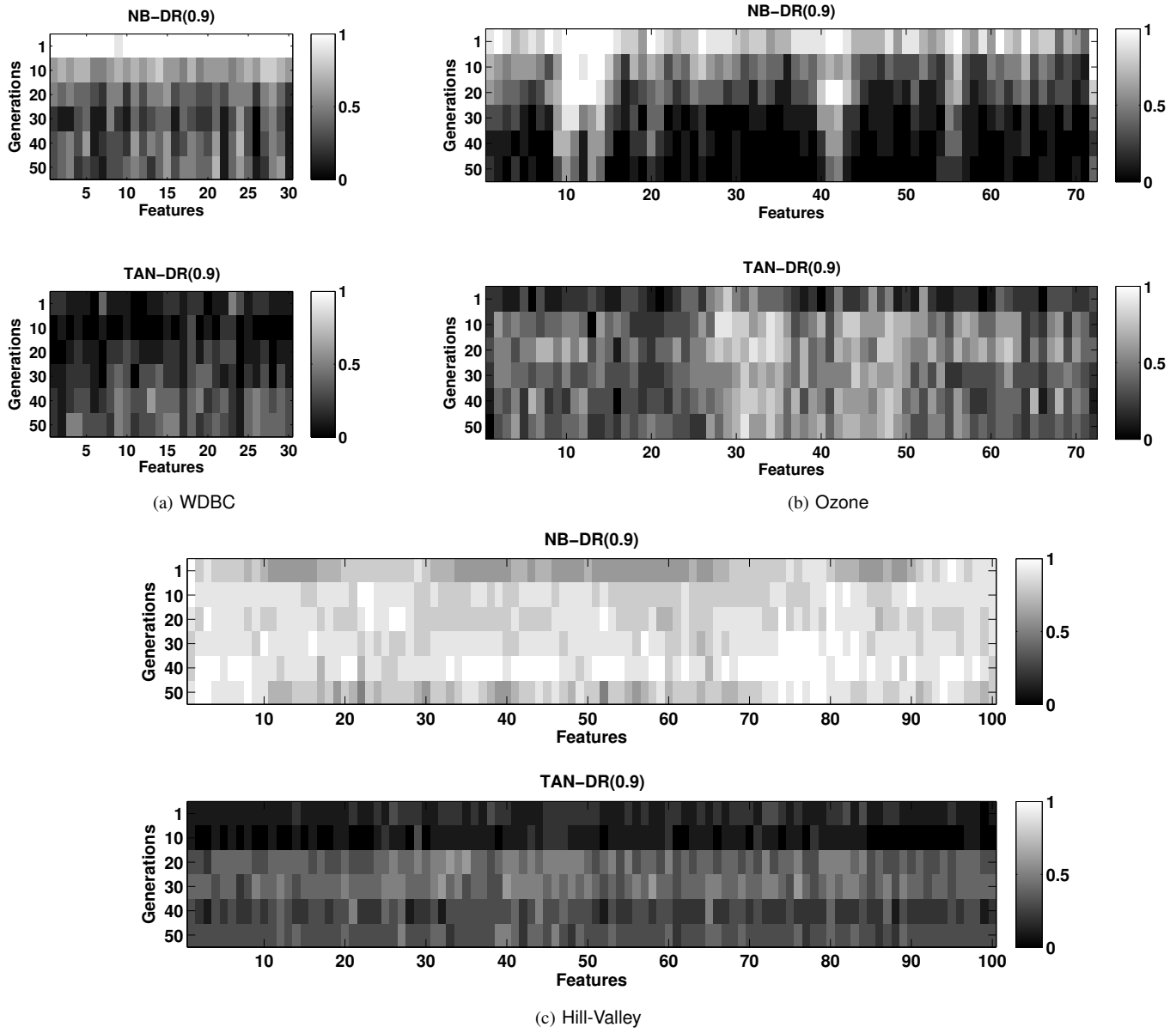
We have presented a multiobjective EDA based on joint modeling of objectives and variables for FSS in classification, although the method can be easily adapted for FSS in clustering. To deal with the inherent noise in the estimation of objective values, they were represented as intervals. The proposed algorithm employs a ranking method that is able to order the solutions in the population when objective values are given as intervals. Based on this ordering, a subset of promising solutions are selected for model estimation.

A twp-step model estimation method was proposed for learning a joint probabilistic model of both objectives and variables from the selected solutions. In the first step, the variable selection property of ℓ_1 regularization technique is used to select a subset of variables with higher relevance to the objectives. This helps to simplify MBN estimation by reducing the space of possible structures and provides an initial approximation of the bridge subgraph structure of MBN. In the second step, a search+score strategy is used to estimate an MBN for the objectives and the set of selected variables. The estimated joint probabilistic model is then used for generating new solutions while taking into account their objective values.

We defined a six objective optimization problem and used the proposed algorithm to select feature subsets for two Bayesian classifiers, NB and TAN, in three different datasets having an increasing number of features. The experimental results show that, although requiring considerably less time, the non-dominated fronts obtained with the proposed ranking method are comparable or better than the fronts found using the well-known PR method. Also, the comparison of results with those of a standard GA showed that the proposed EDA is able to obtain better non-dominated fronts in terms of both convergence and diversity.

The estimated joint probabilistic models were also used to analyze the interactions between objectives and variables. We saw different variable selection behaviors with each of the classifiers which resulted in detecting different patterns of objective interactions. It was observed that, though the level of noise in the objective values is higher when using TAN classifier for solution evaluation, the variable selection method deployed in the first part of the model estimation can help to identify these relationships.

One of the issues that can prevent the wide-spread use of EDAs for discrete optimization problems like FSS is their requirement for large population sizes. Therefore, one of the future lines of research for this work is to add an adaptive method for population size detection, depending on the specifications of the dataset and classifier that define the problem. Another future work is to use the interactions between objectives found in the joint modeling step to improve noise handling in solution ranking.



(a) WDBC

(b) Ozone

(c) Hill-Valley

Fig. 14. The average frequency of selecting variables using RRMs in the first part of joint model estimation, along different generations in 10 independent runs.

REFERENCES

[1] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[2] K. Tan and C. Goh, "Handling uncertainties in evolutionary multi-objective optimization," in *Computational Intelligence: Research Frontiers*, ser. Lecture Notes in Computer Science, J. Zurada, G. Yen, and J. Wang, Eds. Berlin, Heidelberg: Springer, 2008, vol. 5050, pp. 262–292.

[3] P. Larrañaga and J. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[4] J. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, ser. Studies in Fuzziness and Soft Computing. Secaucus, NJ, USA: Springer, 2006, vol. 192.

[5] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra, "Feature subset selection by Bayesian network-based optimization," *Artificial Intelligence*, vol. 123, no. 1-2, pp. 157–184, 2000.

[6] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 245–271, 1997.

[7] S. Davies and S. Russell, "NP-completeness of searches for smallest possible feature sets," in *AAAI Symposium on Intelligent Relevance*. AAAI Press, 1994, pp. 37–39.

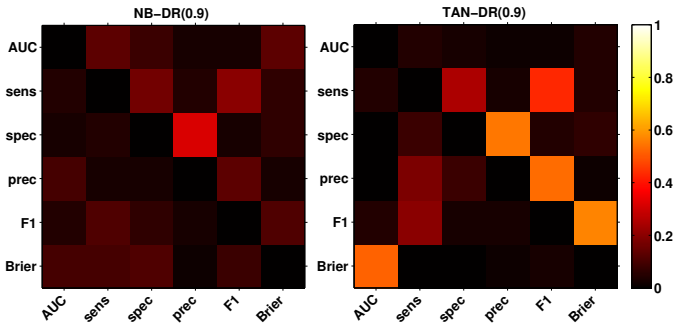
[8] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[9] L. Oliveira, M. Morita, and R. Sabourin, "Feature selection for ensembles using the multi-objective optimization approach," in *Multi-Objective Machine Learning*, ser. Studies in Computational Intelligence, Y. Jin, Ed. Berlin, Heidelberg: Springer, 2006, vol. 16, pp. 49–74.

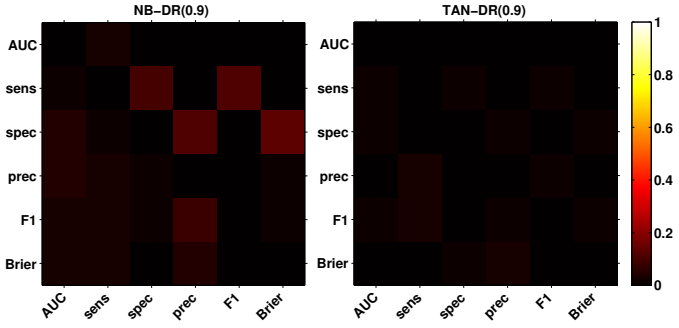
[10] C. Emmanouilidis, A. Hunter, J. MacIntyre, and C. Cox, "A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling," *Journal of Evolutionary Optimization*, vol. 3, no. 1, pp. 1–26, 2001.

[11] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature selection using multi-objective genetic algorithms for handwritten digit recognition," in *16th International Conference on Pattern Recognition (ICPR'02)*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2002, pp. 568–571.

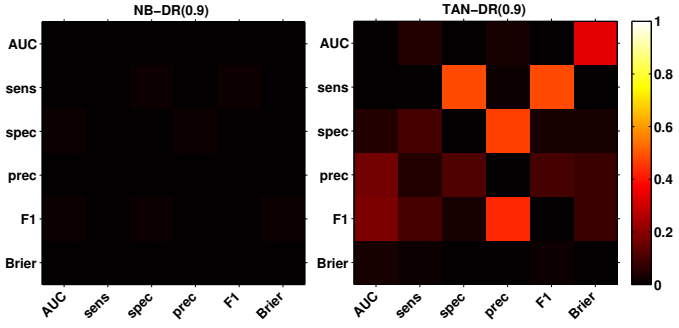
[12] C. Emmanouilidis, A. Hunter, and J. MacIntyre, "A multiobjective evolutionary setting for feature selection and a commonality-based



(a) WDBC



(b) Ozone



(c) Hill-Valley

Fig. 15. The average frequency of arcs in the class subgraph of the MBNs estimated in all generations and in 10 independent runs.

crossover operator," in *IEEE Congress on Evolutionary Computation (CEC'00)*, vol. 1, 2000, pp. 309–316.

[13] J. Horn, N. Nafpliotis, and D. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *First IEEE Conference on Evolutionary Computation (ICEC '94), IEEE World Congress on Computational Intelligence*, vol. 1, 1994, pp. 82–87.

[14] C. Emmanouilidis, "Evolutionary multi-objective feature selection and ROC analysis with application to industrial machinery fault diagnosis," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, ser. EUROGEN 2001, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: International Center for Numerical Methods in Engineering (CIMNE), 2001, pp. 319–324.

[15] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] S. Shi, P. Suganthan, and K. Deb, "Multiclass protein fold recognition using multiobjective evolutionary algorithms," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'04)*, oct. 2004, pp. 61–66.

[18] T. Hamdani, J.-M. Won, A. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *8th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2007)*, ser. Lecture Notes in Computer Science, B. Beliczynski, A. Dzielinski, M. Iwanowski, and B. Ribeiro, Eds., vol. 4431. Berlin, Heidelberg: Springer, 2007, pp. 240–247.

[19] A. Ekbal, S. Saha, and C. S. Garbe, "Feature selection using multiobjective optimization for named entity recognition," in *20th International Conference on Pattern Recognition (ICPR '10)*. IEEE Computer Society, 2010, pp. 1937–1940.

[20] B. Huang, B. Buckley, and T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3638–3646, 2010.

[21] J. Rodríguez and J. Lozano, "Multi-objective learning of multi-dimensional Bayesian classifiers," in *8th International Conference on Hybrid Intelligent Systems (HIS'08)*, 2008, pp. 501–506.

[22] P. V. W. Radtke, T. Wong, and R. Sabourin, "Solution over-fit control in evolutionary multiobjective optimization of pattern classification systems," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 6, pp. 1107–1127, 2009.

[23] Z. Zhu, Y.-S. Ong, and J.-L. Kuo, "Feature selection using single/multi-objective memetic frameworks," in *Multi-Objective Memetic Algorithms*, ser. Studies in Computational Intelligence, C.-K. Goh, Y.-S. Ong, and K. Tan, Eds. Berlin, Heidelberg: Springer, 2009, vol. 171, pp. 111–131.

[24] N. Spolaôr, A. Lorena, and H. Lee, "Multi-objective genetic algorithm evaluation in feature selection," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576. Berlin, Heidelberg: Springer, 2011, pp. 462–476.

[25] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Weihs, "Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 2012, in press, DOI: 10.1007/s00500-012-0874-9.

[26] Y. Kim, W. N. Street, and F. Menczer, "Evolutionary model selection in unsupervised learning," *Intelligent Data Analysis*, vol. 6, no. 6, pp. 531–556, 2002.

[27] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition," in *7th International Conference on Document Analysis and Recognition (ICDAR'03)*, vol. 2. Washington, DC, USA: IEEE Computer Society, 2003, pp. 666–670.

[28] J. Handl and J. Knowles, "Feature subset selection in unsupervised learning via multiobjective optimization," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 217–238, 2006.

[29] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Genetic and Evolutionary Computation Conference (GECCO 01)*. Morgan Kaufmann Publishers, 2001, pp. 283–290.

[30] X. Zhang, B. Lu, S. Gou, and L. Jiao, "Immune multiobjective optimization algorithm for unsupervised feature selection," in *Applications of Evolutionary Computing (EvoWorkshops 2006)*, ser. Lecture Notes in Computer Science, F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. Moore, J. Romero, G. Smith, G. Squillero, and H. Takagi, Eds., vol. 3907. Berlin, Heidelberg: Springer, 2006, pp. 484–494.

[31] D. Zaharie, D. Lungeanu, and S. Holban, "Feature ranking based on weights estimated by multiobjective optimization," in *IADIS European Conference on Data Mining*, J. Roth, J. Gutiérrez, and A. P. Abraham, Eds., 2007, pp. 124–128.

[32] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[33] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *Journal of Heuristics*, vol. 18, no. 5, pp. 795–819, 2012.

[34] D. Thierens and P. A. N. Bosman, "Multi-objective mixture-based iterated density estimation evolutionary algorithms," in *Conference on Genetic and Evolutionary Computation (GECCO '01)*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Francisco, California, USA: Morgan Kaufmann, 2001, pp. 663–670.

[35] M. Pelikan, K. Sastry, and D. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable Optimization via Probabilistic Modeling*, ser. Studies in Computational Intelligence, M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds. Berlin: Springer, 2006, vol. 33, pp. 223–248.

[36] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[37] L. Martí, J. Garcia, A. Berlanga, C. A. Coello Coello, and J. M. Molina, "On current model-building methods for multi-objective estimation of distribution algorithms: Shortcommings and directions for improvement," Department of Informatics, Universidad Carlos III de Madrid, Madrid, Spain, Tech. Rep. GIAA2010E001, 2010.

[38] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multi-objective estimation of distribution algorithm based on joint modeling of objectives and variables," School of Computer Science, Technical University of Madrid, Madrid, Spain, Tech. Rep., 2012.

[39] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[40] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co. Inc., 1989.

[41] J. Teich, "Pareto-front exploration with uncertain objectives," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, and D. Corne, Eds., vol. 1993. Berlin, Heidelberg: Springer, 2001, pp. 314–328.

[42] E. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, and D. Corne, Eds., vol. 1993. Berlin, Heidelberg: Springer, 2001, pp. 329–343.

[43] M. Garza-Fabre, G. Toscano Pulido, and C. Coello Coello, "Ranking methods for many-objective optimization," in *MICAI 2009: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, A. Aguirre, R. Borja, and C. García, Eds. Berlin: Springer, 2009, vol. 5845, pp. 633–645.

[44] ——, "Alternative fitness assignment methods for many-objective optimization problems," in *Artifical Evolution*, ser. Lecture Notes in Computer Science, P. Collet, N. Monmarché, P. Legrand, M. Schoenauer, and E. Lutton, Eds. Berlin: Springer, 2010, vol. 5975, pp. 146–157.

[45] C. Bielza, G. Li, and P. Larrañaga, "Multi-dimensional classification with Bayesian networks," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.

[46] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[47] T. Hesterberg, N. Choi, L. Meier, and C. Fraley, "Least angle and L1 penalized regression: A review," *Statistics Surveys*, vol. 2, no. 2008, pp. 61–93, 2008.

[48] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[49] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the LASSO," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[50] W. Buntine, "Theory refinement on Bayesian networks," in *7th Annual Conference on Uncertainty in Artificial Intelligence (UAI '91)*, B. D'Ambrosio and P. Smets, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1991, pp. 52–60.

[51] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[52] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Fourth International Conference on Parallel Problem Solving from Nature (PPSN IV)*, ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberger, and H.-P. Schwefel, Eds., vol. 1141. Springer, 1996, pp. 178–187.

[53] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the Institute of Radio Engineers*, vol. 49, no. 1, pp. 8–30, 1961.

[54] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.

[55] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[56] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[57] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Master's thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 1995.

[58] S. Bandaru, C. Tutum, K. Deb, and J. Hattel, "Higher-level innovization: A case study from Friction Stir Welding process optimization," in *IEEE Congress on Evolutionary Computation (CEC '11)*, 2011, pp. 2782–2789.

[59] K. Deb, S. Bandaru, and C. Celal Tutum, "Temporal evolution of design principles in engineering systems: Analogies with human evolution," in *Parallel Problem Solving from Nature (PPSN XII)*, ser. Lecture Notes in Computer Science, C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds. Berlin, Heidelberg: Springer, 2012, vol. 7492, pp. 1–10.