Evaluating Ranking Composition Methods for Multi-Objective Optimization of Knowledge Rules

Rafael Giusti Gustavo E. A. P. A. Batista
Ronaldo C. Prati

Institute of Mathematics and Computer Science – ICMC
University of São Paulo – USP
P.O. Box 668, 13560-970, São Carlos, SP, Brazil
{rgiusti, gbatista, prati}@icmc.usp.br

Abstract

Most symbolic classifiers aim at building sets of rules with good coverage and precision. While this is suitable for most applications, they tend to neglect other desirable properties, such as the ability to induce novel knowledge or to show new points of view of well-established concepts. An approach to overcome these limitations involves using a multi-objective evolutionary algorithm to build knowledge rules with specific properties specified by the user. In this paper, we report a research work that combined evolutionary algorithms and ranking composition methods for multi-objective optimization. In this approach, candidate solutions are built, evaluated and ranked according to their performance in each individual objective. Then rankings are composed into a single ranking which reflects the candidate solutions' ability to solve the multi-objective problem considering all objectives simultaneously. We investigate the behavior of 5 *ranking composition methods. These methods are compared and we conclude that all of the studied ranking composition methods provide good balance of objectives. Moreover, for the* 11 *datasets analyzed, we conclude condorcet is the only method which performs statistically better than other methods.*

1 Introduction

Symbolic supervised learning is a field of artificial intelligence that deals with automatic extraction of knowledge from data so that the representation of knowledge is in a form that is intelligible to humans. One of those forms is the knowledge rule, which encodes knowledge in the form of a "`if` *set of conditions* `then` *conclusion*" sentence. A symbolic classifier inducer is a learning algorithm that aims at constructing a set of rules, given some data organized in a set of examples. Usually, symbolic inducers aims at building rule sets with good coverage and precision, which roughly means that they aim at extracting knowledge that applies to a large number of examples and is accurate in classifying new examples. However, typical inducers tend to neglect other desirable properties, such as the ability to induce novel knowledge, to surprise the domain specialist or to show new, possibly contradictory, points of view of well-established concepts.

An approach to overcome these limitations involves using a multi-objective evolutionary algorithm to construct knowledge rules with specific properties specified by the system user [8]. Evolutionary algorithms belong to a family of algorithms that attempt to solve problems by iteratively optimizing a set of sub-optimal solutions. In the case of constructing knowledge rules with specific properties, the sub-optimal solutions are knowledge rules which do not sufficiently satisfy the desired properties. The evolutionary algorithm modifies the sub-optimal rules in order to create new rules that presumably better satisfy those properties. It is important to notice, however, that modifying the content of a rule to improve its ability to satisfy one of the properties may not improve its ability to satisfy the other properties. Moreover, it is often the case that properties are conflicting, i.e., these modifications will decrease the rule's ability to satisfy one or more of the other properties. Thus, the problem of constructing rules with specific properties should be faced as a multi-objective optimization problem where the maximization of each property is one single objective.

Evolutionary algorithms are very suitable for multi-objective optimization. In this paper, we report a research work which combined evolutionary algorithms and ranking composition methods for multi-objective optimization. In this approach, candidate solutions are constructed, evaluated and ranked according to their performance in each individual objective. Then rankings are composed into a single ranking which reflects the candidate solutions' ability to solve the multi-objective problem considering all objectives simultaneously.

In this paper we extend the research in [8] by investigating the behavior of 5 ranking composition methods. These methods are compared and we conclude that all composition methods studied presented similar performance in constructing rules that provide good balance of the measures. Moreover, for the 11 datasets analyzed, we conclude that *condorcet* is the only method which performs statistically better than other methods.

The rest of this paper is organized as follows. Section 2 presents definitions. Section 3 introduces the notion of optimization and ranking composition methods. Section 4 presents related work. Section 5 describes the experiments and results. Finally, Section 6 presents conclusions.

2 Definitions and Notations

A dataset is a set of N examples in the form $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_n)\}$. Each example E_i is a pair $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is a vector of values $(x_{i1}, x_{i2}, .., x_{iM})$ related to a collection of attributes $\{\boldsymbol{X}_1, ..., \boldsymbol{X}_M\}$, whose domains may be finite sets of values (discrete) or infinite intervals of real numbers (continuous). Therefore, the expression x_{ij} refers to the value taken by the attribute $\boldsymbol{X}_j$ in example E_i. The values y_i refers to the special attribute $\boldsymbol{Y}$, called *class* attribute. When all examples have non-empty y_i value, the dataset is said to be *labeled* and supervised learning techniques may be used. This is the case of the research work reported in this paper. Moreover, in this work, the attribute $\boldsymbol{Y}$ takes only discrete values.

When a dataset is labeled and all examples provide a set of discrete values for the class attribute, *classifier inducer* algorithms may be used. A classifier inducer is an algorithm that produces a hypothesis $H : E \rightarrow \boldsymbol{Y}$ which associates an example $e \in E$ to its expected class $y \in \boldsymbol{Y}$. The hypothesis H is called *classifier* and may be represented by a wide variety of description languages. One of the most used is the classification rule representation.

A *classification rule* is a symbolic representation of knowledge, i.e., it is intelligible to humans. A rule R is given in the form $B \rightarrow H$ where B, called *body*, is a conjunction of conditions and H, called *head*, is the y value predicted by the rule. In other words, R *predicts* the class y. Given a rule R and an example e, R *covers* e iff all conditions of $B \in R$ are verified true in e. A rule *correctly* covers an example iff the rule covers the example and correctly predicts its class.

When a rule is evaluated against a dataset, the examples may be distributed along four sets, B, $\overline{B}$, H and $\overline{H}$. Examples covered by a rule belong to B, while examples having the same class as predicted by the rule belong to H. Their complements, $\overline{B}$ and $\overline{H}$ contain the examples *not covered* and the examples *incorrectly* predicted by the rule. The corollary intersections contain examples correctly covered, incorrectly covered, not covered but correctly predicted and not covered but incorrectly predicted. These sets are important to construct the rule's *contingency matrix* (see Table 1), which is the basis of the Lavrač framework [7].

Table 1. The contingency table

	H	$\overline{H}$	
B	f_{hb}	$f_{\overline{h}b}$	f_b
$\overline{B}$	$f_{h\bar{b}}$	$f_{\overline{hb}}$	$f_{\overline{b}}$
	f_h	$f_{\overline{h}}$	1

The Lavrač framework allows one to estimate rule quality levels according to different quality criteria. Assume $X \in \{H, \overline{H}, B, \overline{B}\}$ is the event in which an example is acknowledged as belonging to one of those sets and $x \in \{h, \overline{h}, b, \overline{b}\}$ is the cardinality of those sets (i.e., $h = |H|$ and so forth). If x is interpreted as the number of occurrences of X for a given rule and dataset, one may estimate the probability of the event X, $P(X) = f_x$, based on the frequency of occurrence of X.

One of the most used rule quality measures is *accuracy*, which is defined as the conditional probability of H, given B, $Acc(R) = P(H|B) = \frac{f_{hb}}{f_b}$. Accuracy measures the fraction of examples covered by a rule which have the same class predicted by the rule. However, when the coverage of the rule is low (few examples satisfy the B part of the rule) this probability is likely to be poorly estimated from frequencies. A better estimation is obtained when Laplace correction is used. *Laplace correction* is defined as $LAcc(R) = \frac{hb+1}{b+N_{Cl}}$, where N_{Cl} is the number of the dataset's class values — for the sake of simplicity, intersection operators will be omitted (e.g., $|H \cap B|$ will be written hb).

Support is a widely used rule quality measure. Support is defined as the probability of B and H occurring together, i.e. $Sup(R) = P(HB) = f_{hb}$. Support measures the proportion of examples covered and correctly classified by R.

Novelty is also an important rule quality measure. Novelty is defined as the difference between the probability of B, given H, and the product of the marginal probability of these two events. In general terms, this measure favors slightly inaccurate rules that cover a large number of events more than accurate rules that cover few examples. It can be demonstrated that $|Nov(R)| \leq 0.25$; values near -0.25 indicate strong association between H and B, while values near $+0.25$ indicate strong association between $\overline{H}$ and B. Novelty is defined as $Nov(R) = f_{hb} - f_h \times f_b$.

3 Optimization and Rankings

The construction of knowledge rules with specific properties may be faced as an optimization problem. If there is only one specific property of interest, then the task of optimization is a single-objective optimization problem which requires the maximization or the minimization of a single function. For instance, if one wants to construct rules with high novelty, one may attempt to maximize the function $f(R) = Nov(R)$ which takes a knowledge rule as input and outputs its novelty measure. However, it is most often the case when the user wants to construct rules that present several specific properties, which are very likely to be conflicting. In other words, instead of maximizing one single function, the user is interested in maximizing (or minimizing) several functions. This should be faced as a multi-objective optimization problem and require specific tools to deal with multiple, often contradictory objectives. In fact, even the notion of "optimal" has a different meaning in the case of multi-objective optimization [2]. For instance, in the case of this paper, there is never a single rule that maximizes all desired measures. Instead, we are interested in finding a rule that provide a good balance of the measures instead, even if none of the measures is maximized.

Multi-objective optimization techniques can be divided into two groups. The Pareto-based techniques make use of the Pareto concept [4] and try to find a set of dominant solutions that belong to the Pareto-optimal frontier. The non-Pareto techniques rely in applying some mathematical function to convert a multi-objective instance problem into a single-objective problem. For instance, one may take the weighted mean of the objectives into a function f and apply a single-optimization method to maximize (or minimize) f. This is a simple, elegant approach, but has the severe drawback of only providing good results when an appropriate set of weights is chosen. However, apart from a reduced number of cases where the problem semantics suggest a fairly trivial importance relationship among the objectives, finding an appropriate balance of the weights is a another multi-optimization problem itself.

Ranking composition is a non-Pareto technique that does not require weight balance. A ranking is a collection of items arranged in order according to some quality which they all possess [6]. The position of each item in the ranking is called rank. Ranks are usually expressed as numerical values that reflect their position in the ranking.

Ranking composition is performed in a two-step process, which will be referred to in this paper as the ranking step and the composition step. The ranking step independs of the composition method being used and consists in ranking each item of the collection according to one single objective. The result of this step is a number of rankings that equals the number of optimization objectives, and each item will have the same number of ranks. The next step consists in composing those ranks into a single value, according to the composition method being used. The result of this composition is a final ranking that reveals which item or items are best in providing good balance of all the objectives. It does not reveal, however, *how much* better these items are when compared to the others.

There is not a unique or standard method for the composition step. The most intuitive and straightforward is the *mean* composition, which is the weighted mean of the item's ranks. Given o the number of objectives, $W = \langle w_1, \ldots, w_o \rangle$ a vector of weights related to the objectives and $R = \langle r_1, \ldots, r_o \rangle$ a vector of ranks, the *mean* composition of an item is calculated as $Rk(W,R) = \sum_i \frac{w_i \times r_i}{\Sigma w_i}$. A simple variation of the *mean* composition is the *harmonic* composition, which is obtained by a weighted harmonic mean, instead of an arithmetic mean. *harmonic* composition is given by $Rk(W,R) = \sum_i \frac{\Sigma w_i}{w_i \times r_i^{-1}}$. Another straightforward composition method that is based on simple mathematical functions is the *reciprocal* composition, which is defined as the sum of the inverse of the ranks and is given by the equation $Rk(W,R) = \sum_i \frac{w_i}{r_i}$.

The *median* composition is based on the median statistics of a population. To calculate the median composition of an item's rank, one must first sort its ranks according to the same order criterion used to construct the initial rankings during the ranking step. The item's rank will be the central value of all its ranks — i.e., the median.

The *condorcet* composition is performed in a two-step process. First, all items are compared pairwise on their equivalent ranks. That is, every pair of items (A, B) is compared on their ranks for objective 1, then on their ranks for objective 2 and so forth for all objectives. The item of the pair which "beats" the other for the most number of objectives is considered the winner of the pair and the other item is, logically, the loser. If the items "beat" each other for the same number of ranks or if they have identical ranks for all objectives, then they are tied. The next step involves sorting the items according to their number of wins. If two items have the same number of wins, then their number of loses is used as a second criterion. If two items have the same number of wins and loses, then their number of ties is used as final criterion. This final sorting is a ranking, which should be used as the composition of the initial rankings.

It is important to notice that all composition methods aim at finding good balance of the objectives. However, the notion of "good balance" differs among methods and datasets. Also, not all methods have the same time complexity. We do not provide any formal analysis of the five methods presented, but it is intuitive that all but *condorcet* are $O(n)$ time-complexity, whereas *condorcet* itself is $O(n^2)$, where n is the number of ranked items.

4 Related Work

Rule knowledge extraction with specific properties has received little attention in the past years, as rule learning is most frequently used in the context of classifier induction and association rule learning.

Evolutionary computation has been used in the field for a while to construct knowledge rules or symbolic classifiers. Romao [9] uses genetic algorithms to discover fuzzy rules. Ishibuchi [5] also uses genetic algorithms to construct fuzzy rules, but focusing in evolving classifiers instead of single rules.

In [8], it is proposed the usage of evolutionary algorithms to build individual knowledge rules with specific properties. [8] also presents the Evolutionary Computation Learning Environment — ECLE — as an implementation of that proposition. ECLE is a framework and a learning system which attempts to construct knowledge rules with specific properties out of any initial rule set, which may be composed by randomly generated rules, rules produced by other inducers or rules manually constructed by the system user. This initial rule set is used as the genetic algorithm's initial population.

Genetic algorithms — GA — are iterative optimization algorithms inspired by the theories of natural evolution of the species. GAs work with a set of candidate solutions, called population, which are sub-optimal solutions to the optimization problem. The goal of the GA is to combine and modify those candidate solutions, via crossover and mutation operations, to produce new candidate solutions which are expected to solve the optimization problem more suitably. Both crossover and mutation operators are applied stochastically, in a fashion that resembles the processes of nature. In each iteration, which is called generation, a subset composed by the most suitable rules are selected out of the population. These are the rules that "win" the natural selection process and will "reproduce", that is, will be combined with each other and then modified to compose the new generation. The GA process repeats several times, until an optimal solution is found or another stop criterion is satisfied.

In [8] and in the case of the work reported in this paper, the population was a set of rules which present the specific properties desired by the user, but suboptimally. Ranking composition methods were used to evaluate the rules according to the specific properties and select the rules which would be used to construct the next generation. The convergence criterion was standard deviation convergence. After constructing each population, the algorithm would evaluate the mean of all quality measures. If that statistic's standard deviation was smaller than a threshold value, the GA process was terminated.

5 Experiment Evaluation

In order to evaluate the ranking composition methods discussed in Section 3, several experiments were conducted using 11 datasets from the UCI online dataset repository [1]. Table 2 summarizes the data used in this study. For each dataset, it shows the number of examples (#Examples), attributes (#Attributes), quantitative and qualitative attributes, and majority class frequency.

Table 2. Datasets summary description

Dataset	#Examples	#Attributes (quanti., quali.)	Majority Frequency
Breast	286	9 (9, 0)	70.28%
Bupa	345	6 (6, 0)	57.98%
E.Coli	336	7 (7,0)	89.58%
German	1000	20 (7,13)	70.00%
Glass	214	9 (9,0)	92.06%
Haberman	306	3 (3,0)	73.53%
Heart	270	14 (14,0)	55.60%
New-thyroid	215	5 (5,0)	83.72%
Post-operative	90	8 (1,7)	73.33%
Sonar	208	61 (61,0)	53.50%
Vehicle	846	18 (18,0)	76.48%

For each dataset, a set of rules was generated using classic rule induction algorithms. This set was used as input to a multi-objective optimization algo-

Table 3. Results grouped by dataset and ranking composition method.

Dataset	*condorcet*		*harmonic*		*mean*		*median*		*reciprocal*	
breast	lap:	0.8213	lap:	0.8105	lap:	0.8218	lap:	0.8219	lap:	0.8204
	nov:	0.2655	nov:	0.2422	nov:	0.2622	nov:	0.2709	nov:	0.2580
	sup:	0.4511	sup:	0.4488	sup:	0.4444	sup:	0.4604	sup:	0.4415
bupa	lap:	0.6304	lap:	0.6567	lap:	0.6378	lap:	0.6323	lap:	0.6203
	nov:	0.1507	nov:	0.1402	nov:	0.1437	nov:	0.1550	nov:	0.1601
	sup:	0.2413	sup:	0.2507	sup:	0.2854	sup:	0.2374	sup:	0.1944
ecoli	lap:	0.9711	lap:	0.9711	lap:	0.9710	lap:	0.9716	lap:	0.9711
	nov:	0.2325	nov:	0.2328	nov:	0.2322	nov:	0.2322	nov:	0.2309
	sup:	0.6941	sup:	0.6996	sup:	0.6943	sup:	0.6905	sup:	0.6763
german	lap:	0.8936	lap:	0.8905	lap:	0.8947	lap:	0.8940	lap:	0.8898
	nov:	0.2755	nov:	0.2717	nov:	0.2758	nov:	0.2760	nov:	0.2724
	sup:	0.3139	sup:	0.3126	sup:	0.3131	sup:	0.3142	sup:	0.3141
glass	lap:	0.9564	lap:	0.9466	lap:	0.9558	lap:	0.9553	lap:	0.9560
	nov:	0.0840	nov:	0.0759	nov:	0.0833	nov:	0.0821	nov:	0.0835
	sup:	0.3717	sup:	0.4048	sup:	0.3835	sup:	0.3696	sup:	0.3844
haberman	lap:	0.8135	lap:	0.8107	lap:	0.8098	lap:	0.8096	lap:	0.8099
	nov:	0.2098	nov:	0.2003	nov:	0.2016	nov:	0.2031	nov:	0.1987
	sup:	0.5181	sup:	0.5000	sup:	0.5150	sup:	0.5263	sup:	0.5112
heart	lap:	0.7940	lap:	0.7962	lap:	0.7868	lap:	0.7932	lap:	0.7798
	nov:	0.4573	nov:	0.4108	nov:	0.4401	nov:	0.4502	nov:	0.4363
	sup:	0.3684	sup:	0.3210	sup:	0.3471	sup:	0.3573	sup:	0.3436
new-thyroid	lap:	0.9688	lap:	0.9671	lap:	0.9683	lap:	0.9683	lap:	0.9686
	nov:	0.4491	nov:	0.4424	nov:	0.4471	nov:	0.4477	nov:	0.4468
	sup:	0.7730	sup:	0.7696	sup:	0.7724	sup:	0.7728	sup:	0.7696
post-operative	lap:	0.6121	lap:	0.6242	lap:	0.6168	lap:	0.6147	lap:	0.6137
	nov:	0.1197	nov:	0.1104	nov:	0.1203	nov:	0.1161	nov:	0.1146
	sup:	0.1789	sup:	0.1831	sup:	0.1867	sup:	0.1782	sup:	0.1784
sonar	lap:	0.7199	lap:	0.7312	lap:	0.7200	lap:	0.7162	lap:	0.7235
	nov:	0.3304	nov:	0.3156	nov:	0.3317	nov:	0.3235	nov:	0.3086
	sup:	0.2977	sup:	0.2795	sup:	0.3041	sup:	0.2963	sup:	0.2807
vehicle	lap:	0.9829	lap:	0.9841	lap:	0.9795	lap:	0.9836	lap:	0.9835
	nov:	0.4231	nov:	0.4212	nov:	0.4228	nov:	0.4240	nov:	0.4238
	sup:	0.4701	sup:	0.4666	sup:	0.4758	sup:	0.4699	sup:	0.4699

rithm which used evolutionary computation to construct rules with specific properties out of the initial set of rules. The evolutionary algorithm was a genetic algorithm that combined knowledge by means of two mutation operators and three crossover operators — knowledge rules were combined by having their conditions and parts of their conditions exchanged and individually modified[1]. For this experiment, crossover rate was set to 60% and mutation rate was set to 5%. The task of optimization considered in this study consists of simultaneously maximizing these rules in terms of three measures of rule quality, namely novelty, Laplace and support. These three measures were chosen because they represent some of the most desirable characteristics of knowledge rules discovered by inducers. Rules with high support are applicable to a large number of examples, rules with high Laplace are precise in classifying new examples, and rules with high novelty represent knowledge that is potentially novel to the user or domain expert, even though novelty is a quality that is subjective in nature, and is very difficult to be assessed by an objective measure.

Optimizing simultaneously these three quality measures is not a trivial task. For instance, rules with high support usually are not novel, since knowledge that is applicable to a large number of examples is usually well-known to most domain experts. Similarly, it is usually easier to find precise rules that cover few examples than to find precise rules that cover a large number of examples. Finally, high support rules usually cover a large region in the instance space and have higher probability of false positive classifications.

Experiments were performed with 10-fold cross-validation resampling technique. For each training set, a rule set with all rules induced by $\mathcal{C}4.5$, $\mathcal{C}4.5rules$ and $\mathcal{CN}2$ was given as input to the multi-objective evolutionary algorithm. Although it is not mandatory to input the evolutionary algorithm with a rule set (i.e., the evolutionary algorithm may produce an initial random set of rules), this procedure usually reduces the time required for convergence. Once the convergence condition is satisfied, all optimized rules present in the last generation are evaluated in the training set, and the rule with the best evaluation, i.e., positioned first in the ranking, is selected and evaluated in the test set. In order to improve the statistical confidence, this procedure was repeated 10 times, i.e., 100 train-and-test experiments were performed for each combination of dataset and ranking composition method.

Five ranking composition methods were evaluated: *condorcet*, *median*, *mean*, *reciprocal* and *harmonic*.

[1] See more at [8].

Table 3 shows the average results obtained. Due to lack of space, standard deviations are not reported here[2]. Experimental results show that all composition methods studied present similar performance in constructing rules that provide good balance of the three measures.

In order to analyze whether there is differences among the ranking composition methods, we ran the Friedman test [3]. Friedman test was ran with four different null-hypotheses: (H_1) that the performance of all five methods are comparable considering only the Laplace measure; (H_2) that the performance of all five methods are comparable considering only the novelty measure; (H_3) that the performance of all five methods are comparable considering only the support measure; (H_4) that results obtained using the five ranking composition methods are comparable considering all three measures. When the null-hypothesis is rejected by the Friedman test, we can proceed with a post-hoc test to detect which differences among the methods are significant. We ran the Nemenyi multiple comparison with a control test to point out which methods present a significant difference in the experimental evaluation.

Friedman test results indicate that the null-hypotheses H_1, H_2 and H_3 cannot be rejected. Thus, considering the three quality measures separated, it is not possible to affirm that there is a significant difference among the ranking composition methods with 95% confidence level. However, when all measures are considered together, Friedman test points out that the null-hypothesis H_4 should be rejected. Therefore, we proceeded with Nemenyi to detect which performance differences are significant. The result is pictured in Figure 1. The thicker lines above the graph mark the interval of one critical difference (see [3]) that two methods should have to be statistically different from each other. Groups of algorithms that are not significantly different are also grouped by a line.

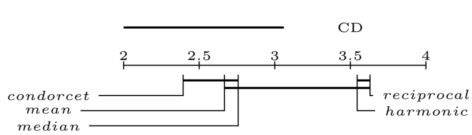

Figure 1. Critical difference diagram for H_4

Figure 1 shows that *condorcet* performs best, followed by *median*, *mean*, *harmonic* and *reciprocal*. Nemenyi test indicated significant difference between *condorcet* and *harmonic* rankings and between *condorcet* and *reciprocal* ranking. There is no significant difference among *condorcet*, *median* and *mean*.

[2] All tabulated results can be found in `http://www.icmc.usp.br/~gbatista/ecle_ranking`.

6 Conclusion

This paper presents a comparison of 5 different ranking composition methods used as part of a genetic algorithm to construct knowledge rules with specific (desired) properties. The performance of this approach was analyzed using 11 datasets from the UCI online repository and three rule quality measures. Experimental results show that all composition methods studied presented similar performance in constructing rules that provide good balance of the three measures. Results also show that there is no statistically significant difference among *condorcet*, *mean* and *median* methods. Moreover, it has been discovered that, for the reported application, the *condorcet* method is statistically better than *harmonic* and *reciprocal* methods.

However, while this approach has provided good results for the presented application, it does not guarantee Pareto-optimality. As future work, we plan to compare this approach with methods based on Pareto-ranking to verify whether there is significant difference of performance.

Acknowledgment: the authors would like to thank the Brazilian funding agency FAPESP for the financial support that made this research work possible.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] C. Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, 2006.

[3] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[4] A. Freitas. A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations*, 6(2):77–86, 2004.

[5] H. Ishibuchi and T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, 141:59–88, 2004.

[6] M. Kendall and J. D. Gibbons. *Rank Correlation Methods*. Edward Arnold, 1948.

[7] N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In *ILP'99*, pages 174–185. Springer-Verlag, 1999.

[8] A. D. Pila, R. Giusti, R. C. Prati, and M. C. Monard. A multi-objective evolutionary algorithm to build knowledge classification rules with specific properties. In *HIS*. IEEE Computer Society, 2006.

[9] W. Romao, A. Freitas, and I. Gimenes. Discovering interesting knowledge from a science & technology database with a genetic algorithm. *Applied Soft Computing*, 4(2):121–137, May 2004.