A Critical Review of Multi-Objective Optimization in Data Mining: a position paper

Alex A. Freitas
University of Kent
Computing Laboratory
Canterbury, CT2 7NF, UK
Tel: 44 1227 82-7220
A.A.Freitas@kent.ac.uk

ABSTRACT

This paper addresses the problem of how to evaluate the quality of a model built from the data in a multi-objective optimization scenario, where two or more quality criteria must be simultaneously optimized. A typical example is a scenario where one wants to maximize both the accuracy and the simplicity of a classification model or a candidate attribute subset in attribute selection. One reviews three very different approaches to cope with this problem, namely: (a) transforming the original multi-objective problem into a single-objective problem by using a weighted formula; (b) the lexicographical approach, where the objectives are ranked in order of priority; and (c) the Pareto approach, which consists of finding as many non-dominated solutions as possible and returning the set of non-dominated solutions to the user. One also presents a critical review of the case for and against each of these approaches. The general conclusions are that the weighted formula approach – which is by far the most used in the data mining literature – is to a large extent an ad-hoc approach for multi-objective optimization, whereas the lexicographic and the Pareto approach are more principled approaches, and therefore deserve more attention from the data mining community.

Keywords

Multi-objective optimization, lexicographic approach, Pareto dominance, Classification.

1. INTRODUCTION

A crucial issue in data mining is how to evaluate the quality of a candidate model – e.g., a classification model such as a rule set or a decision tree. This paper addresses an important aspect of this issue, which is how to evaluate a model's quality by taking into account multiple quality criteria (objectives) to be optimized. In this case the quality of a model can be represented by a n-dimensional vector, where n is the number of quality criteria to be optimized, rather than by a single scalar number. Multi-objective problems are very common in a number of different data mining tasks and problems. As typical examples, let us mention two very generic scenarios, which are broad enough to refer to a large number of data mining projects.

The first scenario involves predictive tasks. Examples of predictive data mining tasks are classification (where the attribute to be predicted is categorical), regression (where the attribute to be predicted is continuous) and dependence modelling (where there are several categorical attributes to be predicted). Here we focus on the classification task, which is in general the most studied in the data mining literature, but the arguments discussed here also hold for other predictive tasks.

The knowledge discovered by a data mining algorithm should be not only accurate but also comprehensible to the user [Fayyad et al. 1996]. Hence, there are a number of classification projects where the goal is to maximize both the predictive accuracy of the classification model and the comprehensibility (simplicity) of the classification model, in order to obtain a model easier to be interpreted by the user. This is the case particularly in the context of decision tree and rule induction algorithms, which lend themselves naturally to the discovery of knowledge in a high-level representation, which can in principle be interpreted by the user – as long as the "complexity" of the model, typically measured by the size of the model, is relatively small. This raises the question of how to evaluate the trade-off between the accuracy of a model and its size.

Indeed, a common approach in the literature consists of reporting the values of both the predictive accuracy and the simplicity (size) of different models produced in a set of experiments [Weiss et al. 2003], [Chisholm & Tadepalli 2002], [Weiss & Indurkhya 2000], [Li et al. 2002]. When reporting these kinds of results, accuracy and simplicity are usually analyzed separately from each other, and it is often the case that "model A is more accurate than model B, but model B is simpler than model A". For instance, which one is a "better" classification model: an easily-interpretable decision tree with a dozen nodes and accuracy rate (on the test set) of 92% or a large, non-interpretable decision tree with hundreds of nodes and accuracy rate of 95%? The answer depends on the problem at hand and the preference of the user.

The second scenario involves attribute selection. This is one of the most studied data preprocessing tasks of the knowledge discovery process, where the goal is to select, out of all original attributes, a subset of attributes that are relevant for the target data mining task [Guyon & Elisseeff 2003]. Again, for the sake of simplicity we focus on attribute selection for the classification task, which is the kind of attribute selection most investigated in the data mining literature, but the arguments discussed here also hold in general for other kinds of data mining tasks, particularly predictive tasks.

In general it is accepted that, in the context of data mining, an attribute selection method should select an attribute subset that not only maximizes the accuracy of the classification model, but also minimizes the number of selected attributes (to save memory space, speed up the classification algorithm, etc.) and/or the size of the classification model built from the selected attributes (to improve the simplicity/interpretability of the model).

Again, a common approach in the literature consists of reporting not only the predictive accuracy of the model built from the selected attributes, but also the number of selected attributes and/or the size of the classification model built from the selected attributes [Yu & Liu 2003], [Kim et al. 2000], [Kohavi & John 1998], [Cherkauer & Shavlik 1996]. It is often the case that "attribute subset A leads to a more accurate model than attribute subset B, but B has a smaller cardinality (number of attributes) than A". This introduces the difficult problem of analyzing the trade-off between maximizing a model's accuracy and minimizing the number of selected attributes, which is analogous to the above mentioned trade-off between a model's accuracy and its size.

We emphasize that the previous two generic scenarios are just a sample of the many kinds of scenario where a data mining problem involves the simultaneous optimization of two or more criteria (objectives). One could easily add to the list scenarios with several other criteria to be optimized. For instance, one could "decompose" the accuracy criterion into the precision and recall criteria, as it is usual in the information retrieval literature [Baeza-Yates & Ribeiro-Neto 1999]. One could measure rule surprisingness (unexpectedness) separately from accuracy and simplicity [Liu et al. 1997], [Freitas 1998]. (The motivation for this is clear by considering the following hypothetical rule: IF (patient is pregnant) THEN (gender is female). This rule is very accurate and very simple, but it gets a very low mark for surprisingness.) One could consider the two objectives of maximizing accuracy and minimizing the cost of the attributes used by the classifier [Turney 1995]. A comprehensive discussion about all kinds of multi-objective data mining scenarios would take a far larger space than the available space, but the above list is, hopefully, enough to show that multi-objective optimization problems are quite common in data mining.

The remainder of this paper is organized as follows. Section 2 discusses three general approaches to cope with multi-objective problems. Section 3 discusses arguments for and against each of those three approaches. Section 4 discusses the relationship between multi-objective optimization and ROC graphs. Finally, section 5 concludes the paper.

2. THREE APPROACHES FOR COPING WITH MULTI-OBJECTIVE PROBLEMS

2.1 The Conventional Weighted-Formula Approach: Transforming a Multi-objective Problem into a Single-objective One

In the data mining literature, by far the most used approach to cope with a multi-objective problem consists of transforming it into a single-objective problem. This is typically done by assigning a numerical weight to each objective (evaluation criterion) and then combining the values of the weighted criteria into a single value by either adding or multiplying all the weighted criteria. That is, the quality Q of a given candidate model is typically given by one of the two kinds of formula:

$$Q = w_1 \times c_1 + w_2 \times c_2 + . . . + w_n \times c_n \quad (1), \text{ or}$$

$$Q = c_1^{W1} \times c_2^{W2} \times . . . \times c_n^{Wn} \quad (2)$$

where w_i, i=1,…n, denotes the weight assigned to criteria c_i, and n is the number of evaluation criteria. Let us mention some examples of this approach in the context of the two multi-objective scenarios discussed in the Introduction.

The first scenario involves rule induction algorithms for classification, where it is common to evaluate the quality of a candidate rule by measuring two or more criteria. An example is:

$$Q = w_1.\text{consistency} + w_2.\text{completeness} \quad (3),$$

an instance of the general formula structure (1) combining completeness and consistency into a single measure of rule quality. This formula and its variations are used in several rule induction algorithms [Bruha & Tkadlec 2003], [Furnkranz & Flach 2003]. Another example is:

$$Q = \text{completeness}^{w} \times \text{consistency\_gain}^{(1-w)} \quad (4),$$

an instance of the general formula structure (2) used in [Kaufmann & Michalski 1999] to produce one of the evaluation criteria used in a lexicographic approach for rule evaluation (to be discussed in section 3.2).

The second scenario involves attribute selection algorithms for classification, where it is also common to evaluate the quality of a candidate attribute subset by measuring two or more criteria. An example is:

$$Q = ¾ \times \text{Acc} + ¼ \times (1 – (S + F)/2) \quad (5).$$

This formula was used by [Cherkauer & Shavlik 1996] to measure the quality of a candidate attribute subset in an attribute selection method following the wrapper approach, where Acc (accuracy) was measured by the validation-set accuracy of a decision tree built with the selected attributes, S was the decision tree size and F was the number of attributes (features) in the candidate attribute subset. Other examples of attribute selection methods following the wrapper approach and combining two or more attribute subset quality criteria in a weighted formula can be found in [Liu & Motoda 1998].

2.2 The Lexicographic Approach

The basic idea of this approach is to assign different priorities to different objectives, and then focus on optimizing the objectives in their order of priority. Hence, when two or more candidate models are compared with each other to choose the best one, the first thing to do is to compare their performance measure for the highest-priority objective. If one candidate model is significantly better than the other with respect to that objective, the former is chosen. Otherwise the performance measure of the two candidate models is compared with respect to the second objective. Again, if one candidate model is significantly better than the other with respect to that objective, the former is chosen, otherwise the performance measure of the two candidate models is compared with respect to the third criterion. The process is repeated until one finds a clear winner or until one has used all the criteria. In the latter case, if there was no clear winner, one can simply select the model optimizing the highest-priority objective.

A well-known algorithm using this approach is the AQ18 rule induction algorithm [Kaufmann & Michalski 1999] and its variants. AQ18 uses a "lexicographic evaluation functional (LEF)" defined by a sequence of pairs <(c_1,t_1), (c_2,t_2), …, (c_n,t_n)>, where each c_i, i=1,…,n, represents the value of a performance criterion for a given candidate model, and each t_i, i=1,…,n, represents the tolerance associated with c_i. This tolerance is specified by a threshold indicating the maximum value that the performance criterion c_i for a given candidate model is allowed to deviate from the value of c_i for the best current candidate model. More precisely, let M_1 and M_2 be two candidate models being compared, and assume, without loss of generality, that M_1 has a better value of c_i than M_2. If the difference between M_1's c_i value and M_2's c_i value is greater than t_i, then M_1 is immediately considered better than M_2, without the need to check lower-priority objectives. Otherwise the difference between M_1 and M_2 is not considered significant, and in order to select the best candidate model one has to use the remaining lower-priority objectives. In [Kaufmann & Michalski 1999] this approach is used with a LEF where a predictive accuracy-related measure (a combination of completeness and consistency gain) is used as the highest-priority criterion (c_1), and rule description simplicity as the next criterion (c_2); but of course other criteria could be used.

2.3 The Pareto Approach

The basic idea of the Pareto approach is that, instead of transforming a multi-objective problem into a single-objective problem and then solving it by using a single-objective search method, one should use a multi-objective algorithm to solve the original multi-objective problem. Intuitively, this approach makes sense. One should adapt the algorithm to the problem being solved, rather than the other way around. In any case, this intuition needs to be presented in more formal terms, which is done in the following.

Let us start with a definition of Pareto dominance. A solution s_1 is said to dominate (in the Pareto sense) a solution s_2 if and only if s_1 is strictly better than s_2 with respect to at least one of the criteria (objectives) being optimized and s_1 is not worse than s_2 with respect to all the criteria being optimized. Mathematically, assuming – without loss of generality – that all criteria c_i, i=1,…k, are to be maximized, a solution s_1 dominates a solution s_2 if and only if $\exists c_i$ such that $s_1(c_i) > s_2(c_i)$ and $\forall c_i$, i=1,…k, $s_1(c_i) \geq s_2(c_i)$, where $s_1(c_i)$ denotes the quality of solution s_1 with respect to criteria c_i and k is the number of criteria being optimized. A solution s_i is said to be non-dominated if and only if there is no solution s_j that dominates s_i. Note that the Pareto approach never mixes different criteria into a single formula – all criteria are treated separately.

The concept of Pareto dominance is illustrated in Figure 1, where the two objectives to be maximized are the accuracy and the simplicity of a classification model. In the figure, model A is dominated by model B and by model D; model C is dominated by model D; and model E is dominated by model F. Models B, D and F are non-dominated solutions. They form the so-called Pareto front.

Once Pareto dominance has been defined, the next step is to understand the crucial differences between a multi-objective algorithm based on Pareto dominance and a single-objective algorithm. There are two related crucial differences. First, there is a difference in the kind of output expected from each of these two kinds of algorithm. A multi-objective algorithm should return to the user a *set* of non-dominated solutions, rather than just a *single* solution as in a single-objective algorithm.

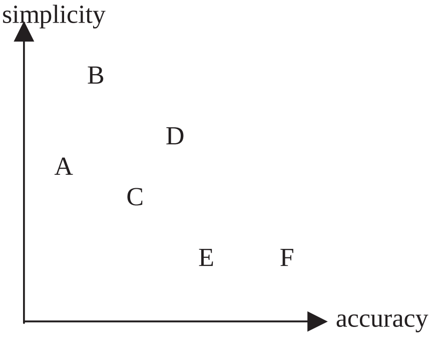

Figure 1: Examples of Pareto dominance

Second, and related to the first difference, the search performed by a multi-objective algorithm should explore a considerably wider area of the search space and keep track of all non-dominated solutions found so far, in order to find as many solutions in the Pareto front as possible. Of course, this makes a Pareto-based multi-objective optimization algorithm more complex than its single-objective counterpart. Some Pareto-based multi-objective optimization data mining algorithms are discussed in [Kim et al. 2000], [Bhattacharyya 2000], [Pappa et al. 2002].

3. ARGUMENTS FOR AND AGAINST EACH MULTI-OBJECTIVE OPTIMIZATION APPROACH

3.1 Arguments For and Against the Conventional Weighted-Formula Approach

Argument For – Simplicity

This approach, discussed in section 2.1, has the advantage of conceptual simplicity and easy of use, which probably explains its popularity. However, it has several drawbacks, which are discussed in the following.

Argument Against – The "magic number" problem and missed opportunities

The most obvious problem with weighted formulae such as formulas (1) and (2) is that, in general, the setting of the weights in these formulas is ad-hoc, based either on a somewhat vague intuition of the user about the relative importance of different quality criteria or in trial and error experimentation with different weight values. In other words, each weight seems a "magical number", which often is justified in the literature with suitably vague sentences such as "the values of these weights were empirically determined".

Another problem with weights is that, once a formula with precise values of weights has been defined and given to a data mining algorithm, the data mining algorithm will be effectively trying to find the best model for that particular setting of weights, missing the opportunity to find other models that might be actually more interesting to the user, representing a better trade-off between different quality criteria. In particular, weighted formulas involving a linear combination of different quality criteria have the limitation that they cannot find solutions in a non-convex region of the Pareto front. This point will be discussed in section 3.3.

For now, to see the limitation of linear combinations of different criteria, consider a hypothetical scenario where we have to select the best candidate to the position of data miner in a company, taking into account two criteria c_1 and c_2 – say, their amount of knowledge about machine learning and statistics, measured by a test and/or a detailed technical interview. Suppose the first candidate's scores are $c_1 = 9.5$ and $c_2 = 5$; the second candidate's scores are $c_1 = 7$ and $c_2 = 7$; and the third candidate's scores are $c_1 = 5$ and $c_2 = 9.5$. The choice of the "best" candidate should depend, of course, on the relative importance assigned to the two criteria by the employer. It is interesting to note, however, that although it is trivial to think of weights for criteria c_1 and c_2 that would make the first or third candidate the winner, it is actually impossible to choose weights for c_1 and c_2 so that the second candidate would be the winner – assuming that the weighted formula is a *linear* combination of weights, such as formula (1). Intuitively, however, the second candidate might be the favorite candidate of many employers, since she/he is the only one to have a good knowledge about both machine learning and statistics.

Of course, it is possible to make the second candidate the winner. One has to use a non-linear combination of criteria, e.g. the formula $c_1 \times c_2$ – an instance of the generic formula (2), where the exponent weights are set to 1. However, in this example one would have chosen the formula $c_1 \times c_2$ *a posteriori*, i.e., after one has learned that the user would prefer the second candidate, rather than the other two candidates. If one had asked a search algorithm to find the best candidate without this a posteriori knowledge about the candidates, one could easily have provided the algorithm with a formula involving a linear combination of weights (which are more often used in the literature than non-linear formulas), and in this case one would miss the opportunity of finding a candidate such as the second above candidate. In other words, it is hard for the user to define the best setting of weights *a priori*, without knowing the results of the research.

Argument Against – Mixing different units of measurement

This problem is particularly serious when the weighted formula involves a summation/subtraction (rather than a multiplication/division) of terms representing different quality criteria, such as formulas (1) and (5). Different model-quality criteria often have very different scales in their units of measurement. For instance, accuracy and classification model-size (e.g. the number of decision tree nodes) are measured in very different scales. This problem can be dealt with by normalizing the different quality criteria so that they refer to the same scale. This approach is well-known in the literature and at first glance it is a very satisfactory approach. There is, however, a subtle problem associated with normalization that is rarely discussed in the literature. In essence, the problem is that in general there are several different ways of normalizing a quality measure, and the decision about which normalization procedure should be applied tends to be ad-hoc. This problem can also be understood in the context of inductive biases. An inductive bias is any criterion (explicit or implicit), except consistency with the data, used to favour one hypothesis over another [Mitchell 1980], [Mitchell 1997]. A normalization procedure is a source of inductive bias, in the sense that it tends to favour one hypothesis over others and it is not based on consistency with the data – assuming that a predictor attribute is normalized without taking into account the class values, which is usually the case. It is well-known that an inductive bias has a domain-dependent effectiveness, so that any inductive bias will be suitable for some application domains and unsuitable for others.

To see an example of the subtle problem associated with normalization, consider the previously-mentioned work of [Cherkauer & Shavlik 1996] on attribute selection, where the quality of a candidate attribute subset was evaluated by formula (5). In that formula, the value of S (decision tree size) was normalized by dividing the tree size by the number of training examples the decision tree was built from. Consider now an alternative normalization (not used in the paper): dividing the current tree size by the size of the largest tree among all the trees generated by the search so far. Both normalization procedures produce a value in the range 0..1, as desired, but the two procedures might produce quite different values of S to be used in formula (5) – which should influence the choice of the weight values in formula (5). Which of them is the "best" normalization? One cannot mathematically prove that one of them is always the best, in the same sense that one cannot mathematically prove that an inductive bias is always the best. Hence, ideally the choice of the "best" normalization procedure should involve background knowledge about the application domain and/or trial and error experimentation with different normalization procedures and weight values. This tends to be an ad-hoc approach, rather than a principled approach.

Argument Against – Mixing "apples and oranges" (non-commensurable criteria)

This is another subtle problem associated with the weighted-formula approach, which is often ignored in the literature. Before discussing this problem in the context of measuring the quality of a model, let us explain the core of the problem by using some simple examples and common-sense arguments.

Common sense tells us that non-commensurable criteria should not be added/subtracted to/from each other in a formula. For instance, if the salary of a customer is US$ 50,000 and the number of dependents (e.g. children) of the customer is 5, it does not make sense to add 50,000 + 5. This would produce a meaningless quantity. Note that the problem here is not only the fact that the two attributes being added have very different scales. This problem can be solved by normalizing both attributes into a range 0..1, say by dividing the salary and the number of dependents by the maximum values of these attributes among the customers in the database. This problem was discussed in the previous item. In this item we are interested in another problem. Even after performing normalization, we still have the problem that salary and number of dependents are non-commensurable criteria. In other words, they measure very different attributes of a customer, and the addition/substraction of these values in a weighted formula does not make any sense at all, regardless of normalization. It would produce a quantity that would be meaningless to the user, which would go against a basic goal of data mining, namely that discovered knowledge should be ultimately understandable to the user [Fayyad et al. 1996].

At first glance it could be argued that this is a problem only if the different criteria are added/subtracted, but not if they are multiplied/divided. Indeed, in some cases the multiplication/division of different criteria produces perfectly meaningful attributes. In the previous example, one can divide the salary of a customer by her/his number of dependents, which intuitively would produce a meaningful indicator if we were trying to classify customers into, say, "good credit" and "bad credit" customers. However, even when using a formula involving multiplication/division, the produced quantity may not be meaningful to the user in many cases. For instance, dividing (or multiplying) salary by age does not appear to produce a very meaningful indicator to a user.

Let us now turn to the problem of mixing non-commensurable criteria in a weighted formula evaluating a candidate model. In particular, let us consider the problem of mixing accuracy and comprehensibility (simplicity) measures into the same formula, since these are probably the two model-quality criteria most used in data mining. Clearly, accuracy and comprehensibility are two very different, non-commensurable criteria to evaluate the quality of a model. Actually, comprehensibility is an inherently subjective, user-dependent criterion. Even if we replace the semantic notion of comprehensibility by a syntactic measure of simplicity such as model size, as it is usually done when evaluating "comprehensibility", the resulting measure of simplicity is still non-commensurable with a measure of accuracy. The crucial problem is not that these two criteria have different units of measurement (which can be, to some extent, "reasonably solved" by normalization, as discussed earlier), but rather that they represent very different aspects of a model's quality. In principle, it does not make sense to add/subtract accuracy and simplicity, and the meaningfulness of an indicator multiplying/dividing these two quality criteria is questionable. A more meaningful approach is to recognize that accuracy and simplicity are two very different quality criteria, and treat them separately, without mixing them in the same formula.

3.2 Arguments For and Against the Lexicographic Approach

Argument For – Recognizing the non-commensurability of different quality criteria

The lexicographic approach has one important advantage over the weighted-formula approach: the former avoids the problem of mixing non-commensurable criteria in the same formula. Indeed, the lexicographic approach treats each of the criteria separately, recognizing that each criterion measures a different aspect of quality of a candidate solution. As a result, the lexicographic approach avoids the three drawbacks associated with the weighted-formula approach discussed in section 3.1 – namely the "magic number" problem, the problem of mixing different units of measurement and the problem of mixing "apples and oranges".

In addition, although the lexicographic approach is somewhat more complex than the weighted-formula approach, the former can still be considered conceptually simple and easy to use. In particular, the lexicographic approach is considerably simpler and easier to use than the Pareto approach.

Argument Against – Introducing a new Ad-Hoc Parameter

As discussed earlier, the lexicographic approach usually requires one to specify a tolerance threshold for each criterion. It is not trivial how to specify these thresholds in a principled manner. A commonplace approach is to use a statistics-oriented procedure, e.g. standard deviation-based thresholds, which allow us to reject a null hypothesis of insignificant difference between two objective values with a certain degree of confidence. Although this approach is statistically sound, it should be recalled that it still requires the specification of one parameter, the degree of confidence. This specification still has a certain degree of arbitrariness, since any "high enough" value such as 95% or 99% could be used. Of course one can always ask the user to specify the thresholds or any other parameter, but this introduces some arbitrariness and subjectiveness in the lexicographic approach – analogous to the usually arbitrary, subjective specification of weights for different criteria in the weighted formula approach.

3.3 Arguments Against and For The Pareto Approach

Argument against – Multiple runs of a single-objective optimization algorithm seems "enough"

It could be argued that we do not need Pareto-based multi-objective optimization because a data mining algorithm based on the conventional weighted formula approach can be used to analyse the trade-offs associated with different criteria to be optimized. One simply has to run the algorithm multiple times, with a different set of weights in each run. However, this argument does not seem very convincing, for the reasons discussed in the next item.

Argument for – Multiple runs of a single-objective optimization algorithm is inefficient and ineffective

First, it should be noted that multiple runs of a single-objective optimization algorithm is an ad-hoc approach, since there is no principled, mathematically-sound method to decide which weights should be used in each run, nor to decide how many runs of the algorithm should be performed. Second, this is an inefficient approach [Deb 2001], [Corne et al. 2003], because each run will be effectively ignoring the candidate solutions evaluated by the previous runs of the algorithm, so that later runs can spend a considerable time re-evaluating some solutions that had already been evaluated by earlier runs. Third, this is an ineffective approach. There is no mechanism to enforce the desired property that the solutions discovered by the different runs should be as spread as possible along the Pareto front. In addition, no matter how many times we run a conventional weighted-formula algorithm with a linear weighted formula (which is the most used kind of weighted formula), it will never find a non-dominated solution in a non-convex region of the Pareto front, such as the solutions indicated by the black circle in Figure 2. The figure assumes – without loss of generality – that both objectives are to be minimized. (See also Section 3.1, first "Argument against" heading.)

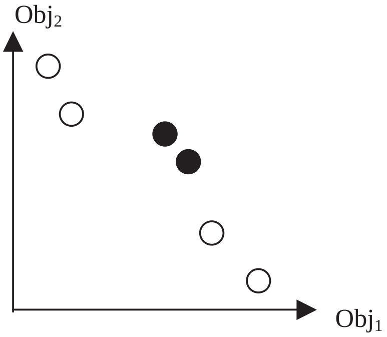

Figure 2: Example of non-dominated solutions in the non-convex region of the Pareto front (shown in black)

Argument against – The Minimum Description Length Principle seems "enough"

It could be argued that, in problems where the objectives to be optimized are a model's accuracy and size, we do not need Pareto-based multi-objective optimization, because we already have the Minimum Description Length Principle. This principle is often used to favour the discovery of knowledge that is both accurate and simple [Tuzhilin 2002], [Quinlan & Rivest 1989], [Fayyad & Irani 1993]. In essence, this principle recommends that, given a set of competing hypotheses (predictive models for a given data set), one should choose as the "best" hypothesis the one that minimizes the sum of two terms, namely: (a) the length of the hypothesis; and (b) the length of the data given the hypothesis, i.e., the length of the data when encoded using the hypothesis as a predictor for the data. The second term represents the length of the encoding of the data instances that are "exceptions" to the hypothesis. One characteristic of this principle is that both terms (a) and (b) are measured in bits. At first glance, this has the nice characteristic that two terms that seemed non-commensurable (accuracy and size of the hypothesis) have been transformed into a common unit of measurement, and have therefore become "commensurable".

Although the Minimum Description Length Principle seems an elegant solution for obtaining commensurability between accuracy and simplicity, it introduces another problem, as discussed in the next item.

Argument for – The Minimum Description Length Principle comes with a price

The previously-mentioned commensurability is artificially obtained at the following price: the MDL principle introduces the problem of how to encode a hypothesis and its data exceptions into bits of information. For any hypothesis space with a reasonably large size, there will be a large number of different ways of encoding hypotheses and exceptions in bits of information. Finding a "good" encoding scheme, among so many possible encoding schemes, is usually a very difficult task – see e.g. [Quinlan & Rivest 1989] – and the value of the previously mentioned terms (a) and (b) is entirely dependent on the choice of encoding scheme. Actually, one cannot say that one encoding is superior to others in general, because each encoding is associated with an inductive bias, and it is well-known that the effectiveness of any inductive bias is application dependent. Hence, in general the choice of the encoding to be used is difficult and typically done *manually*, either taking into account background knowledge or in a more ad-hoc fashion involving trial and error – a situation that is somewhat analogous to the manual choice of the weights for each objective in the weighted-formula approach.

Pareto-based optimization avoids the problems associated with the choice of a good encoding, since it treats the model-quality criteria of accuracy and size as two separate quality measures that are never mixed into the same formula, respecting the natural non-commensurability of these two quality measures.

In addition, note that the Pareto approach is more generic than the Minimum Description Length principle, since the latter is used only to cope with accuracy and simplicity, whereas the Pareto approach can cope with any kind of non-commensurable model-quality criteria.

Argument against – The difficulty of choosing a single "best" solution to be used in practice

A possible criticism of the Pareto approach is that in this approach the data mining algorithm returns a set of non-dominated solutions, whereas in practice the user will often use a single solution. How can one choose the "best" non-dominated solution, out of all non-dominated solutions? This seems a difficult problem associated with the Pareto approach.

Arguments for – A better role for the user or a data-driven criterion to choose the "best" solution

There are at least two possible answers to the question posed by the previous item. The first answer to the problem of choosing a single "best" solution, which is the answer most commonly found in the multi-objective optimization literature, it that it is up to the user to choose the best solution, by taking into account her/his background knowledge and preferences. At first glance, this might seem a not very satisfactory answer, because it introduces some subjectivity into the choice of the best solution. However, this problem is actually less serious than it looks like at first glance, and this criticism can be rebutted by two points.

First, one should recall that data mining is just one step of a broader knowledge discovery process, and this process is highly interactive [Fayyad et al. 1996], [Brachman & Anand 1996] because participation of the user in the process is essential to improve the chance that discovered knowledge will be actually useful for the user. Second, and this is a subtle point that is often missed by critics of the Pareto approach, the conventional approach for coping with multi-objective problems – viz., using a weighted formula – is also associated with an important *subjective* decision, namely the choice of the weight values for each of the different criteria.

With respect to the latter point, the difference between the weighted formula approach and the Pareto approach can be summarized by the two flowcharts in Figures 3(a) and 3(b). This Figure clearly shows that in both approaches the user must make a subjective decision. The difference is that in the weighted-formula approach the user has to make the subjective decision about the weight values *a priori*, before the data mining algorithm is run. Intuitively, this is a very uninformed decision, because at this point the user does not have any computational result to support her/him in the task of analysing the trade-offs associated with the different criteria. By contrast, in the Pareto approach the user makes a subjective choice about the best classifier *a posteriori*, after she/he has seen the several non-dominated solutions returned by the data mining algorithm. Those solutions cover a wide range of different trade-offs between the criteria being optimized, so that in principle the user is now in a much better position to analyze the trade-offs associated with the different criteria and choose the best solution by taking into account her/his preferences.

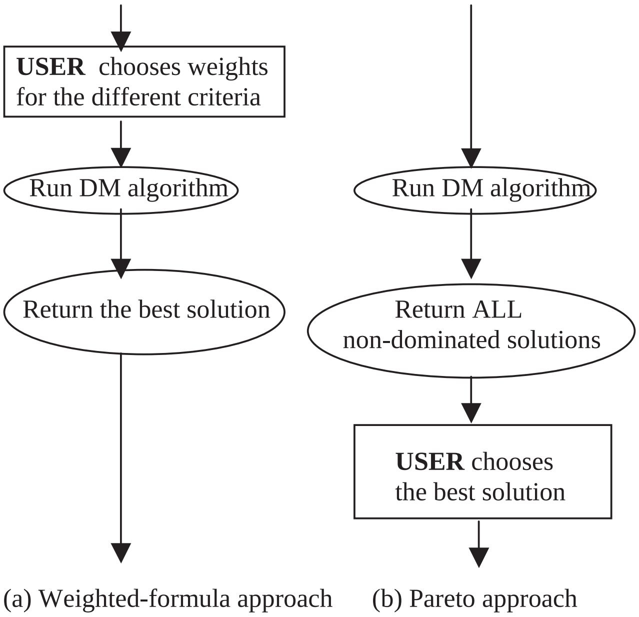
USER chooses weights for the different criteria
Run DM algorithm
Return the best solution
Run DM algorithm
Return ALL non-dominated solutions
USER chooses the best solution
(a) Weighted-formula approach
(b) Pareto approach

Figure 3: Two approaches for solving multi-objective optimization problems

In any case, there is also a second answer to the problem of how to choose the "best" solution out of all the non-dominated solutions. One can actually violate the principle of returning all non-dominated solutions and return only the "best" non-dominated solution found by the algorithm, according to a data-driven heuristic used by the algorithm. One possibility is as follows. Although all non-dominated solutions, by definition, have in common the characteristic of not being dominated by any other solution, they are still different from each other with respect to the number of solutions that they dominate. That is, different non-dominated solutions will dominate a different number of (dominated) solutions. This suggests the use of a quality measure that can be used as a "tie-breaking criterion" to select the "best" non-dominated solution, out of all non-dominated solutions. In essence, one can choose the non-dominated solution that dominates the largest number of (dominated) solutions among all solutions generated by the algorithm [Deb 2001].

It should be noted that this approach is a significant departure from the "conventional" Pareto approach represented by Figure 3(b). At first glance, one might criticize this approach as an unnecessarily complex way of implementing a single-objective optimization algorithm, since just a single "optimal" solution is returned anyway. However, recall that in this approach the algorithm is still performing a multi-objective search, looking for a diverse set of non-dominated solutions spread across the Pareto front. The wider exploration of the search space could still be very beneficial, by finding a non-dominated solution that could not be explored by a conventional single-objective algorithm using, say, the weighted formula approach.

4. THE RELATIONSHIP BETWEEN MULTI-OBJECTIVE OPTIMIZATION AND ROC GRAPHS

ROC graphs are an increasingly popular way of analyzing the performance of a classifier, and they are particularly useful to choose the best classifier under different scenarios of class distribution and misclassification costs [Provost & Fawcett 1997], [Ting 2002]. On a ROC graph, the false positive rate (FPR) is plotted on the X axis and the true positive rate (TPR) is plotted on the Y axis. The performance of classifiers can then be visualized on this graph, as follows. Some classifiers produce a binary output – a positive or negative class. These classifiers are represented by points in the ROC graph. Other classifiers produce a numeric output, to which a threshold is applied in order to determine if the predicted class is positive or negative. These classifiers are represented by curves in the ROC graph – corresponding to a continuous series of (FPR, TPR) pairs as the threshold values are varied. A ROC graph is illustrated in Figure 4. In a ROC graph, the ideal performance corresponds to the upper-left point (0,1), and the strategy of randomly guessing the class of an example corresponds to the line y = x, shown as the dashed line in the figure.

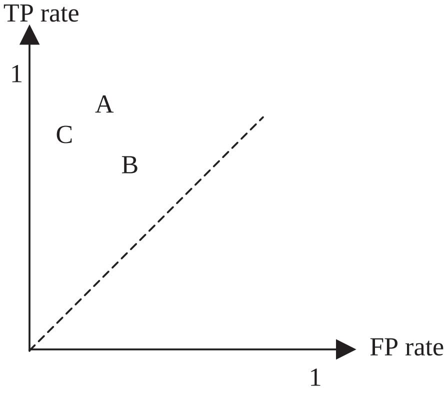

Figure 4: a ROC graph

Let us consider first the case of binary classifiers. A given point A in the ROC graph is better than another point B if A is to the northwest of B, that is, if A is better than B with respect to at least one of the two criteria (lower FPR, higher TPR) and A is not worse than B with respect to any of the two criteria. If these conditions hold, then A dominates B, i.e., A is better than B across all class and misclassification cost distributions. An example is shown in Figure 4, where classifier A dominates classifier B. Note that this is precisely the definition of Pareto dominance, i.e., a given point A in the ROC graph is better than another point B across all class and misclassification cost distributions if and only if A dominates B in the Pareto sense. In the case of numeric classifiers, a given classifier A is better than another classifier B only when the entire curve of A is to the northwest of B. This is clearly a generalization of Pareto dominance from individual points to continuous curves. In addition, a classifier is potentially optimal if and only if it lies on the northwest boundary of the convex hull [Provost & Fawcett 1997]. Note that this boundary is precisely the Pareto front in the context of a multi-objective optimization problem – involving the minimization of FPR and the maximization of TPR.

As mentioned earlier, it is clear that if a classifier A dominates a classifier B then A is a better classifier. However, it is often the case that one classifier obtains better performance than another only in a limited range of the ROC graph, corresponding to a limited range of class and misclassification cost distributions. This is what happens, for instance, with classifiers A and C in Figure 4. Both are non-dominated classifiers, lying in the "Pareto front" of the ROC graph. In this case, it has been suggested that one should choose the classifier that has the highest value of the area under the ROC curve (AUC) [Ling & Zhang 2002], [Bradley 1997]. AUC is a single measure summarizing the performance of a classifier across the entire range of class distribution and misclassification costs. However, since the AUC is just a global measure summarizing performance across a wide range of scenarios, it will lead to the choice of a suboptimal classifier in many scenarios [Provost & Fawcett 1997]. Hence, as pointed out by Provost & Fawcett, it is important to discover classifiers that are spread across the northwest boundary of the convex hull, which corresponds to discover classifiers along the Pareto front. This is clearly an argument supporting multi-objective optimization algorithms, whose goal is precisely to discover solutions spread across the Pareto front.

To summarize, *ROC Graph-based analysis of classifier performance can be regarded as a particular case of the more general principle of Pareto dominance*.

5. CONCLUSIONS

This paper has presented a critical review of three different approaches for coping with multi-objective problems in data mining, namely: (a) the "conventional" approach of transforming a multi-objective problem into a single-objective one via a weighted formula; (b) the lexicographical approach; and (c) the Pareto approach. The weighted formula approach is by far the most popular approach in the data mining literature. However, a careful analysis of this approach has revealed its drawbacks. One of these drawbacks is that it mixes different non-commensurable model-quality criteria into the same formula. In particular, this has the disadvantage of producing model-quality measures that are not very meaningful to the user, going against the principle that in data mining discovered knowledge should be not only accurate but also comprehensible to the user [Fayyad et al. 1996].

This and other drawbacks of the weighted-formula approach are avoided by the lexicographic approach and the Pareto approach. The latter two approaches are more principled approaches for coping with multi-objective optimization problems in data mining.

With respect to the influence of the user in the result returned by the data mining algorithm, the lexicographic approach can be considered as an intermediary approach between the weighted-formula and the Pareto approach, as follows. As discussed earlier, in the weighted-formula approach the user has the full responsibility for specifying the weights (defining the relative importance of each of the objectives to be optimized) a priori, before she/he has knowledge about the solution candidates to be explored by the algorithm. By contrast, in the Pareto approach the user does not need to specify any weight nor any other form of assigning different priority to different objectives. The algorithm will search for all non-dominated solutions, implicitly considering that all objectives have "the same priority", and only after the set of non-dominated solutions is returned by the algorithm the user will have to choose one particular solution, a posteriori. Finally, the lexicographic approach allows the user to assign different priorities to different objectives in a kind of *qualitative* fashion, i.e., the user just has to say, for instance, "objective A is more important than objective B", without having to specify the precise *quantitative* value of weights for objectives A and B. Hence, the user's task becomes considerably simpler.

In any case, the lexicographic approach shares with the weighted-formula formula the characteristic that a single solution is returned to the user. This has the advantage of simplicity but the disadvantage of missed opportunities (i.e., the user misses the opportunity of analysing the trade-off of different non-dominated solutions) discussed earlier.

The Pareto approach is sometimes criticized as being "unnecessarily complex". However, in the discussion presented in section 3.3 several arguments against the Pareto approach were found wanting and were rebutted by counter-arguments. It is true that the Pareto approach is considerably more complex (in terms of designing and running the data mining algorithm) than the other two approaches. However, this disadvantage seems compensated by its several advantages – in particular, avoiding multiple runs of the algorithm, avoiding ad-hoc specification of parameters and returning to the user a very informative set of non-dominated solutions.

The Pareto approach's property of returning a set of non-dominated solutions not only offers the user a rich source of information about the trade-offs between different model-quality criteria, but also has other potential applications in data mining. In particular, the diversity of solutions in the Pareto front naturally lends itself to the use of those solutions in the creation of an ensemble of models (e.g., classifiers). Several techniques for generating an ensemble of models rely on a kind of randomization – e.g. random subsets of data. By contrast, the Pareto approach offers the interesting alternative of performing an explicit search for diverse, non-dominated models, and all the non-dominated discovered models can be immediately used to compose a diverse ensemble of models.

Of course, multi-objective optimization is *not* a panacea. There are scenarios where a problem that is, at first glance, a multi-objective problem can be effectively casted as a single-objective problem. An example is the "two-objective" problem where a company wants to identify customers satisfying two criteria, viz.: a) having a high probability of churning, and b) representing a high revenue for the company. At first glance this might look like a two-objective scenario. However, in this scenario the two objectives are very related, and they can be effectively transformed into a single objective by multiplying them, which will compute the expected revenue loss due to churning.

However, it should be noted that in these scenarios, although a multi-objective optimization algorithm might not be necessary, it might still offer some benefits associated with the fact that its search considers a diverse set solutions. For instance, [Bhattacharyya 2000] addressed a problem where a cellular-phone provider wanted to identify churners satisfying the two above-mentioned criteria. The author developed a Pareto-based multi-objective optimization method to solve the two-objective optimization problem, obtaining good results – outperforming more conventional single-objective algorithms.

In any case, we emphasize that, similarly to several other techniques in data mining, the effectiveness of different multi-objective optimization approaches strongly depends on the application domain. For instance, recall that the lexicographic approach requires the user to specify a priority ordering for the objectives. This can be natural and desirable in some applications, but may be unnatural or undesirable in other applications. In addition, the computational complexity associated with the Pareto approach might make it cumbersome in some applications.

As a general conclusion of the discussion presented in this paper, both the lexicographic approach and the Pareto approach are more principled approaches to cope with multi-objective data mining problems than the conventional weighted-formula approach. Hence, the former two approaches deserve more attention from the data mining community. In particular, much more work is needed to compare these three approaches, both empirically (in a large number of different data sets and different scenarios) and theoretically, since projects comparing two or more multi-objective approaches are rare in the data mining literature.

6. REFERENCES

[Baeza-Yates & Ribeiro-Neto 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[Bhattacharyya 2000] S. Bhattacharyya. Evolutionary algorithms in data mining: multi-objective performance modelling for direct marketing. *Proc. 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-2000)*, 465-473. ACM, 2000.

[Bradley 1997] A.P. Bradley. The use of the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition 30(7)*, 1145-1159. 1997.

[Brachman & Anand 1996] R.J. Brachman and T. Anand. The process of knowledge discovery in databases: a human-centered approach. In: U.M. Fayyad et al (Eds.) *Advances in Knowledge Discovery and Data Mining*, 37-58. AAAI/MIT, 1996.

[Bruha & Tkadlec 2003] I. Bruha and J. Tkadlec. Rule quality for multiple-rule classifier: empirical expertise and theoretical methodology. *Intelligent Data Analysis 7(2)*, 2003, 99-124.

[Cherkauer & Shavlik 1996] K.J. Cherkauer and J.W. Shavlik. Growing simpler decision trees to facilitate knowledge discovery. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, 315-318. AAAI Press, 1996.

[Chisholm & Tadepalli 2002] M. Chisholm and P. Tadepalli. Learning decision rules by randomised iterative local search. *Proc.19th Int. Conf. on Machine Learning (ICML-2002)*, 75-82. Morgan Kaufmann, 2002.

[Corne et al. 2003] D. Corne, K. Deb and P.J. Fleming. The good of the many outweighs the good of the one: evolutionary multi-objective optimization. *IEEE Connections Newsletter 1(1)*, 9-13. *IEEE Neural Networks Society*, Feb. 2003.

[Deb 2001] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[Fayyad & Irani 1993] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributges for classification learning. *Proc. 13th Int. Joint Conf. on Artificial Intelligence (IJCAI'93)*, 1022-1027. 1993.

[Fayyad et al. 1996] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth. From data mining to knowledge discovery: an overview. In: U.M. Fayyad et al (Eds.) *Advances in Knowledge Discovery and Data Mining*, 1-34. AAAI/MIT, 1996.

[Freitas 1998] A.A. Freitas. On objective measures of rule surprisingness. *Principles of Data Mining & Knowledge Discovery (Proc. 2nd European Symp., PKDD'98). LNAI 1510*, 1-9. Springer-Verlag, 1998

[Furnkranz & Flach 2003] J. Furnkranz and P.A. Flach. An analysis of rule evaluation metrics. *Proc. 20th Int. Conf. on Machine Learning (ICML-2003)*. Morgan Kaufmann, 2003.

[Guyon & Elisseeff 2003] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research 3 (2003)*, 1157-1182. 2003.

[Kaufmann & Michalski 1999] K.A. Kaufmann and R.S. Michalski. Learning from inconsistent and noisy data: the AQ18 approach. *Foundations of Intelligent Systems (Proc. ISMIS-99). LNAI 1609*, 411-419. Springer, 1999.

[Kim et al. 2000] Y. Kim, W.N. Street and F. Menczer. Feature selection in unsupervised learning via evolutionary search. *Proc. 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-2000)*, 365-369. ACM, 2000.

[Kohavi & John 1998] R. Kohavi and G.H. John. The wrapper approach. In: H. Liu and H. Motoda (Eds.) *Feature Extraction, Construction and Selection: a data mining perspective*, 33-50. Kluwer, 1998.

[Li et al. 2002] J. Li, R. Topor, H. Shen. Construct robust rule sets for classification. *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD-2002)*. ACM Press, 2002.

[Ling & Zhang 2002] C.L. Ling and H. Zhang. Toward Bayesian classifiers with accurate probabilities. *Advances in Knowledge Discovery & Data Mining (Proc. PAKDD-2002), LNAI 2336*, 123-134. Springer, 2002.

[Liu & Motoda 1998] H. Liu and H. Motoda (Eds.) *Feature Extraction, Construction and Selection: a data mining perspective*. Kluwer, 1998.

[Liu et al. 1997] B. Liu, W. Hsu and S. Chen. Using general impressions to analyze discovered classification rules. *Proc. 3rd Intl. Conf. on Knowledge Discovery & Data Mining (KDD-97)*, 31-36. AAAI Press, 1997.

[Mitchell 1980] T.M. Mitchell. The need for biases in learning generalizations. Rutgers Technical Report, 1980. Published in: J.W. Shavlik and T.G. Dietterich (Eds.) *Readings in Machine Learning*, 184-191. Morgan Kaufmann, 1990.

[Mitchell 1997] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Pappa et al. 2002] G.L. Pappa, A.A. Freitas and C.A.A. Kaestner. A multiobjective genetic algorithm for attribute selection. Proc. 4th Int. Conf. on Recent Advances in Soft Computing (RASC-2002), 116-121. Published in CD-ROM (ISBN: 1-84233-0764). Nottingham Trent University, UK. Dec. 2002.

[Provost & Fawcett 1997] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD-97)*, 43-48. AAAI Press, 1997.

[Quinlan & Rivest 1989] J.R. Quinlan and R.L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation 80*, 227-248. 1989.

[Ting 2002] K.M. Ting. Issues in classifier evaluation using optimal cost curves. *Proc.19th Int. Conf. on Machine Learning (ICML-2002)*, 642-649. Morgan Kaufmann, 2002.

[Turney 1995] P. D. Turney. Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. *J. AI Research 2,* Mar. 1995, 369-409.

[Tuzhilin 2002] A. Tuzhilin. Minimum Description Length. In: W. Klosgen & J.M. Zytkow (Eds.) *Handbook of Data Mining and Knowledge Discovery*, 490-496. Oxford University Press, 2002.

[Weiss & Indurkhya 2000] S.M. Weiss and N. Indurkhya. Lightweight rule induction. *Proc.17th Int. Conf. on Machine Learning (ICML-2000)*, 1135-1142. Morgan Kaufmann, 2000.

[Weiss et al. 2003] S.M. Weiss, S.J. Buckley, S. Kapoor and S. Damgaard. Knowledge-based data mining. *Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-2003)*, 456-461. ACM, 2003.

[Yu & Liu 2003] L. Yu and H. Liu. Efficiently handling feature redundancy in high-dimensional data. *Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-2003)*, 685-690. ACM, 2003.

About the author:

Dr. Alex Freitas earned a Ph.D. in Computer Science from the University of Essex, UK, in 1997. He is currently a Lecturer at the University of Kent, UK. His publications include two authored books about data mining, several invited book chapters and more than 70 refereed papers published in journals or conferences. His main research interests are data mining, biologically-inspired algorithms and bioinformatics.