Cost Sensitive Discretization of Numeric Attributes

Tom Brijs[1] and Koen Vanhoof[2]

[1]Limburg University Centre, Faculty of Applied Economic Sciences, B-3590 Diepenbeek, Belgium
tom.brijs@rsftew.luc.ac.be

[2] Limburg University Centre, Faculty of Applied Economic Sciences, B-3590 Diepenbeek, Belgium
koen.vanhoof@rsftew.luc.ac.be

Abstract. Many algorithms in decision tree learning are not designed to handle numeric valued attributes very well. Therefore, discretization of the continuous feature space has to be carried out. In this article we introduce the concept of cost sensitive discretization as a preprocessing step to induction of a classifier and as an elaboration of the error-based discretization method to obtain an optimal multi-interval splitting for each numeric attribute. A transparant description of the method and steps involved in cost sensitive discretization is given. We also evaluate its performance against two other well known methods, i.e. entropy-based discretization and pure error-based discretization on a real life financial dataset. From the algoritmic point of view, we show that an important deficiency from error-based discretization methods can be solved by introducing costs. From the application point of view, we discovered that using a discretization method is recommended. To conclude, we use ROC-curves to illustrate that under particular conditions cost-based discretization may be optimal.

1 Introduction

Many algorithms which focus on learning decision trees from examples are not designed to handle numeric attributes. Therefore, discretization of continuous valued features must be carried out as a preprocessing step. Many researchers have already contributed to the issue of discretization, however as far as we know, no efforts have been made to include the concept of misclassification costs to find an optimal multi-split. Discretization also has some additional appeals. Kohavi & Sahami [1996] mentioned that discretization itself may be considered as a form of knowledge discovery in that critical values in a continuous domain may be revealed. Catlett [1991] also reported that, for very large data sets, discretization significantly reduces the time to induce a classifier.

Traditionally, five different axes can be used to classify the existing discretization methods: *error-based* vs. *entropy-based*, *global* vs. *local*, *static* vs. *dynamic*, *supervised* vs. *unsupervised* and *top-down* vs. *bottom-up*. Our method is an error-based, global, static, supervised method combining a top-down and bottom-up

approach. However, our method is not just an error-based method. Through the introduction of a misclassification cost matrix, candidate cutpoints are evaluated against a cost function to minimize the overall misclassification cost of false positive and false negative errors instead of just the total sum of errors. False positive (resp. false negative) errors, in our experimental design, are companies incorrectly classified as not bankrupt (bankrupt) although actually they are bankrupt (not bankrupt).

The objective of this paper is to evaluate the performance of our cost sensitive discretization method against Fayyad & Irani's entropy-based method. First, we evaluate the effectiveness of both methods in finding the critical cutpoints that minimize an overall cost function as a result of the preprocessing knowledge discovery step. Secondly, both methods will be compared after induction of the C5.0 classifier to evaluate their contribution to decision tree learning.

In section 2, we introduce the concept of cost sensitive discretization and a transparant description of the several steps that have been undertaken to achieve cost sensitive discretization will be given. In section 3, we elaborate on related work in the domain of discretization of continuous features. In section 4, an empirical evaluation of both methods is carried out on a real life dataset. Section 5 is reserved for a summary of this work.

2 Cost Sensitive Discretization

Cost sensitive discretization signifies taking into account the cost of making errors instead of just minimizing the total sum of errors. This implies that discretizing a numeric feature involves searching for a discretization of the attribute value range that minimizes a given cost function. The specification of this cost function is dependent on the costs assigned to the different error types. Potential interval splittings can then be generated and subsequently evaluated against this cost function. To illustrate the process of cost sensitive discretization we consider a hypothetical example of a numeric attribute for which 3 boundary points are identified (see figure 1). In each interval the number of cases together with their class labels are given. Intuitively, a boundary point is a value V in between two sorted attribute values U and W so that all examples having attribute value U have a different class label compared to the examples having attribute value W, or U and W have a different class frequency distribution. Previous work [Fayyad and Irani 1992] has contributed substantially in identifying potential cutpoints. They proved that it is sufficient to consider boundary points as potential cutpoints, because optimal splits always fall on boundary points. Formally, the concept of a boundary point is defined as:

Definition 1 [Fayyad and Irani 1992] *A value T in the range of the attribute A is a boundary point iff in the sequence of examples sorted by the value of A, there exist two examples* $s_1, s_2 \in S$, *having different classes, such that* $val_A(s_1) < T < val_A(s_2)$; *and there exists no other example* $s' \in S$ *such that* $val_A(s_1) < val_A(s') < val_A(s_2)$.

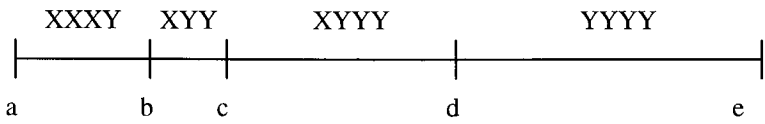

Fig. 1. an example (numeric-valued) attribute and its boundary points.

Now suppose C(Y | X) = 3, i.e misclassifying a class X case as belonging to class Y costs 3 and C(X | Y) = 1. For a real life dataset these cost parameters may be found in the business context of the dataset. However, in many cases, exact cost parameters are not known. Usually the cost parameters of true positive and true negative classifications are set to null but the cost-values for false positive and false negative errors reflect their relative importance against each othter. This results in what is called a cost matrix. The cost matrix indicates that the cost of misclassifying a case will be a function of the predicted class and the actual class. The number of entries in the cost matrix is dependent on the number of classes of the target attribute. Consequently, for each potential interval the minimal cost can be calculated by multiplying the false positive cost (respectively false negative cost) by the false positive (respectively false negative) errors made as a result of assigning one of both classes to the interval and picking the minimal cost of both assignments. For example, the total minimal cost for the overall interval (from a to e) is 10 which can be found as follows:

the number of X cases in interval a-e is 5
the number of Y cases in interval a-e is 10, so
classifying all cases in a-e as 'X' gives a cost of 10 * C(X | Y) = 10 * 1 = ***10***
classifying all cases in a-e as 'Y' gives a cost of 5 * C(Y | X) = 5 * 3 = 15
therefore, the minimal cost for the overall interval a-e is ***10***.

Suppose the maximum number of intervals k is set to 3, now a network can be constructed as depicted in figure 2 (not all costs are included for the sake of visibility). The value of k may be dependent on the problem being studied, but Elomaa & Rousu [1996] advise to keep the value of k relatively low. Increasing parameter k reduces the misclassification cost after discretization but it has a negative impact on the interpretability of the classification tree after induction because the tree will become wider.

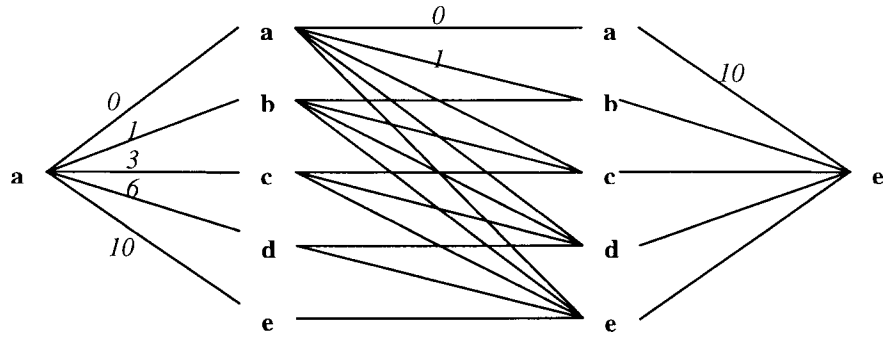

Fig. 2 Shortest route network.

The shortest route linear programming approach can be used to identify the optimal number and placement of the intervals that yields the overall minimal cost for the discretization of this numeric attribute.

This short illustration shows that several phases are to be undertaken to discretize numeric attributes in a cost sensitive way. First, all boundary points for the attribute under consideration must be identified. Second, a cost matrix must be constructed to specify the cost of making false positive and false negative errors. Third, costs must be calculated and assigned to all potential intervals. Fourth, the maximum number of intervals k must be specified. Fifth, a shortest route network can be constructed from all potential intervals with their corresponding minimal costs. Finally, this shortest route network can be solved using the shortest route linear programming routine.

3 Related Work

Traditional error-based methods, for example Maas [1994], evaluate candidate cutpoints against an error function and explore a search space of boundary points to minimize the sum of false positive and false negative errors on the training set. Entropy-based methods, for example Fayyad and Irani [1993], use entropy measures to evaluate candidate cutpoints. Our method is an *error-based* discretization method. However, through the introduction of a misclassification cost matrix, candidate cutpoints are evaluated against a cost function to minimize the overall cost of false positive and false negative errors instead of just the total sum of errors. Kohavi and Sahami [1996] show that the error-based discretization method has an important deficiency, i.e. it will never generate two adjacent intervals when in both intervals a particular class prevails even when the class frequency distributions differ in both intervals. Our cost sensitive discretization method however does not suffer from this deficiency because it takes into account these class frequency differences. By increasing the error-cost of the minority class, the frequency of the minority class is leveraged so that, eventually, different class labels will be assigned to both intervals, indicating a potential cutpoint.
For previous work on other discretization methods we refer to [Holte 1993: static discretization] versus [Fulton, Kasif & Salzberg 1994: dynamic discretization] and [Van de Merckt 1993: unsupervised discretization] versus [Holte 1993 or Fayyad and Irani 1993: supervised discretization] and [Fayyad and Irani 1993: top-down discretization] versus [Kerber 1992: bottom-up discretization].

4 Empirical Evaluation

4.1 The Data Set

To carry out our experiments we used a real life dataset of 549 tuples, all tuples representing a different company. Each company is described by 18 continuous valued attributes, i.e. different financial features on liquidity, solvability, rentability, and others. The entire set is not included because of space limitations. The target attribute is a 2-class nominal attribute indicating whether the company went bankrupt (class 0) or not (class 1) during that particular year. The class distribution in this data

set is highly unbalanced, containing only 136 (24.77%) companies that went bankrupt. This data set was gathered from the official financial statements of these companies which are available at the National Bank of Belgium. In Belgium medium to big enterprises are obliged to report their financial statements in large detail.

4.2 Position of Cutpoints and Induced Cost

In a first experiment we evaluated the performance of Fayyad & Irani's discretization method against our cost sensitive method relative to a specified cost function. We know our method yields optimal results, i.e it achieves minimal costs given a maximum of k intervals. Now, we are interested if Fayyad & Irani's entropy-based method is able to come close to this optimal solution by using the same cost function. We discretized all numeric attributes seperately for different misclassification cost matrices ranging from false positive cost parameter 1 (uniform cost problem) to 8 (false positive errors are severely punished relative to false negative errors). For the sake of simplicity we call this cost parameter the *discretization cost*. Parameter k was arbitrarily set to $2n+1$, i.e 5, with n the number of classes. We are particularly interested in to what extent our cost sensitive method is able to achieve significantly better results on the overall cost function compared to the method of Fayyad & Irani. Our experiments revealed that for all attributes, cost sensitive discretization achieved significantly better results than Fayyad & Irani's entropy-based discretization. On average, for all discretization costs and for all attributes, entropy-based discretization resulted in a 8.7 % increase in cost relative to our method. Table 1 shows the percentage increase in cost of Fayyad & Irani's method against our method for different discretization costs.

Table 1 Average percentage increase in cost Fayyad vs. cost sensitive

Discretization Cost	1	2	3	4	5	6	7	8
	6.3	6.9	7.1	7.1	8.7	9.0	11.4	13.1

This large performance gap is mainly due to the fact that our cost sensitive method exploits local class differences to achieve lower costs whereas the entropy-based method finds thresholds to minimize the entropy function and, as a consequence, it is not so heavily distracted by local differences.

Experiments revealed that on average entropy-based discretization results in fewer cutpoints (2) compared to cost-sensitive discretization (4). For low false positive costs, cost-sensitive discretization only subdivides the range where the frequency of the minority class equals that of the majority class or small frequency differences exist, resulting in sensitivity to local class frequency differences. For high false positive costs only the common range with high class frequency differences is subdivided.

4.3 Comparison of entropy-based versus cost-based discretization

False positive(FP) and false negative(FN) error rates We induced the C5.0 classifier and used repeated 3-fold cross validation on the discretized data sets to

compare the FP and FN error rate for both discretization methods. The C5.0 algorithm has been used with increasing FP cost. Increasing this cost results in a lower FP error rate and a higher FN error rate. In order to visualize the differences between the different discretization methods, we normalized the error rates and used entropy-based discretization as the base line. This means that a given percentage is the cost sensitive error rate divided by the entropy-based error rate. First, we want to investigate the interaction effect of a given discretization cost and changing the C5.0 cost parameter on the FP and FN error rates. When the discretization cost equals 1, the method is an error-based method. On figure 3 it can be seen that for this method the FP error rate is higher but the FN error rate is lower than the base line (entropy-based).

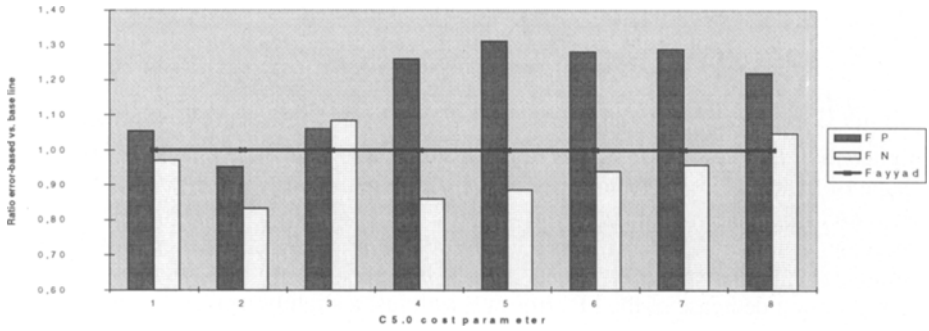

Fig. 3 FP and FN error rates discretization cost 1 vs. Fayyad & Irani

This higher number of FP errors results from the fact that some attribute value ranges are highly dominated by non-bankruptcy cases and thus will be classified as non-bankrupt while this range can still contain a large proportion of bankruptcy cases. Calculation of the frequency of FP errors (43.6%) confirmed this observation. With the given class distribution, the global accuracy is higher with the error based discretization. The following figures (4 and 5) give the error rates for other discretization costs (resp. 2 and 6).

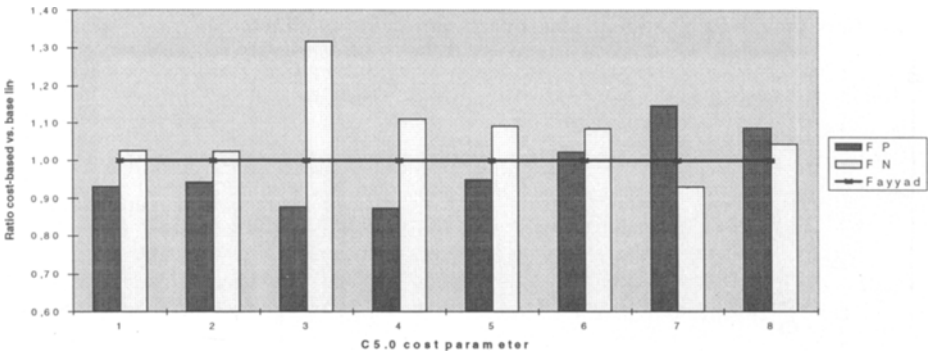

Fig. 4 FP and FN error rates discretization cost 2 vs. Fayyad & Irani

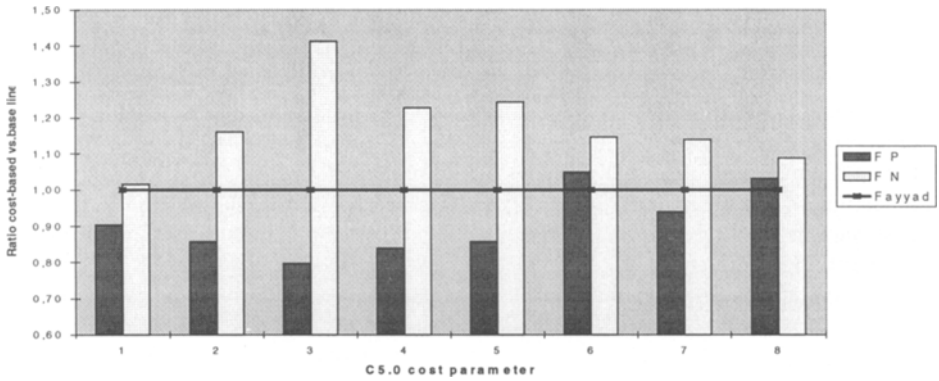

Fig. 5 FP and FN error rates discretization cost 6 vs. Fayyad & Irani

For the cost sensitive discretization the FP error rate is lower and the FN error rate is higher than the base line. These figures show two surprising results. Firstly, the shape of the curves. By increasing the C5.0 cost parameter, the FP error rate should decrease. This is the case for both methods, but for the cost sensitive discretizer the decrease is much faster (at lower C5.0 cost parameter). For higher C5.0 cost parameters, the entropy-based method has a lower FP error rate. Secondly, the first bar(s) show a decrease of the FP error rate with just a slightly increase in the FN error rate. Combining these two results indicates that the best results can be obtained by using a low C5.0 cost parameter.

Misclassification costs We will compare the performance of the classifiers obtained with the different discretization methods in a ROC graph [Provost & Fawcett 1997]. On a Roc graph the true positive rate (TP) is plotted on the Y axis and the false positive rate (FP) on the X axis. One point in the ROC curve (representing one classifier with given parameters) is better than another if it is to the north-west (TP is higher, FP is lower or both). A ROC graph illustrates the behaviour of a classifier without regard to class distributions and error cost, so that it decouples classification performance from these factors. Figure 6 shows the ROC graph for the different sets of classifiers. We decided not to show all classifiers, only the most relevant classifiers with respect to the performance evaluation are shown.

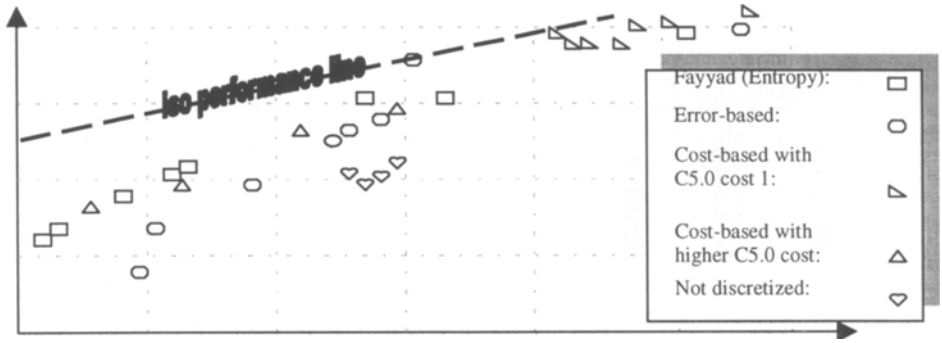

Fig. 6 Roc curve for different classifiers

Provost has shown that a classifier is potentialy optimal if and only if it lies on the northwest boundary of the convex hull [Barber, Dobkin and Huhdanpaa 1993] of the set of points in the ROC curve. From the figure we can see that the set of classifiers with the entropy-based and cost-based discretization method are potentialy optimal. From a visual interpretation, we can rank the methods for the region with a FP rate lower than 35 as follows: entropy, better than cost-based, better than error-based discretization. For the region with a FP rate higher than 35: firstly cost-based and secondly entropy and error-based discretization. To choose the optimal (minimal misclassification cost) classifier we need to know the error cost and the class distributions. In Belgium, the FP error cost is estimated to be 30 to 50 times higher than the FN error cost and the prior probability (this is the true distribution) of negative classes versus positive classes is estimated to be approximately 1 in 95. With this information a set of iso-performance lines [Provost & Fawcett 1997] with a slope of 30/95 can be constructed. On an iso-performance line all classifiers corresponding to points on the line have the same expected cost and the slope of the line is dependent on the a priori cost and class frequency distributions. This provides us with an instrument to choose the optimal classifier of the given sets of classifiers. If only the single best classifier is to be chosen, under the known cost and class frequency distributions, the error-based classifier (indicated by a circle) slightly outperforms the cost-based discretization methods with C5.0 cost 1 (indicated by a right triangle), as can be seen on figure 6. Altogether, it can be seen on the same figure that discretization of numeric attributes prior to induction is always better than discretizing while inducing the C5.0 classifier.

Overfitting With our dataset, by using a discretization method, better estimated accuracies are obtained due to the fact that overfitting is reduced. When using the C5.0 cost functionality this observation is even strengthened. Increasing the C5.0 cost parameter results in increasing resubstitution error rates as illustrated in table 2. The overfitting pattern is similar to that of the false negative error rates shown in the first paragraph of section 4.3. From these observations, it can also be seen that cost sensitive discretization is less robust compared to entropy-based discretization, but more robust than C5.0 without discretization. So, cost sensitive discretization is more able to lower the false positive error rate but is more sensitive to overfitting than entropy-based discretization.

Table 2 Overfitting in absolute percentages

C5.0 cost parameter	1	2	3	4	5	6	7	8
Not discretized	11.96	15.06	13.68	15.07	14.48	13.88	12.87	11.93
Fayyad	10.95	12.31	10.69	9.77	9.45	8.53	8.00	7.83
Average of all cost sensitive	9.16	11.75	12.54	11.39	10.94	10.48	9.88	9.42

5 Conclusion

The concept of misclassification costs is an important contribution to the work of error-based discretization because in many real world problems, the cost of making certain mistakes is not equal. As a consequence, false positive and false negative

classifications were treated equally. A discretization method that is cost sensitive has been implemented, tested and compared with a well known discretization method on a real life financial dataset. From an algoritmic point of view, it has been shown that an important deficiency from error-based discretization methods can be solved by introducing costs, i.e. two adjacent intervals with different class labels can be generated even when in both intervals a particular class prevails. From the application point of view, we may conclude that using a discretization method is recommended. C5.0 is overfitting the financial dataset. It is easier to reduce this overfitting by a priori discretization than by tuning the C5.0 pruning parameter. Choosing the optimal discretization method is more difficult. Firstly, the results are only valid for this small dataset. Secondly, dependent on the evaluation procedure and distributions used, different choices are possible. The three methods considered are all potentialy optimal, but cost sensitive discretization of numeric attributes has showed to be worth considering further research.

References

1. Barber C., Dobkin D., and Huhdanpaa H. (1993). The quickhull algorithm for convex hull. Technical Report GCG53, University of Minesota.
2. Catlett J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the Fifth European Working Session on Learning*, 164-178. Berlin: Springer-Verlag.
3. Dougherty J., Kohavi R., and Sahami M. (1995). Supervised and unsupervised discretization of continous features. In *Machine Learning: Proceedings of the Twelfth Int. Conference*, 194-202. Morgan Kaufmann.
4. Elomaa T., and Rousu J. (1996). Finding Optimal Multi-Splits for Numerical Attributes in Decision Tree Learning. Technical Report NC-TR-96-041, University of Helsinki.
5. Fayyad U., and Irani K. (1992). On the handling of continuous-valued attributes in decision tree generation. In *Machine Learning 8*. 87-102.
6. Fayyad U., and Irani K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth Int. Joint Conference on Artificial Intelligence*, 1022-1027. Morgan Kaufmann.
7. Fulton T., Kasif S., and Salzberg S. (1995). Efficient algorithms for finding multi-way splits for decision trees. In *Proceedings of the Twelfth Int. Conference on Machine Learning*, 244-251. Morgan Kaufmann.
8. Holte R. (1993). Very simple classification rules perform well on most commonly used datasets. In *Machine Learning 11*, 63-90.
9. Kerber R. (1992). Chimerge: Discretization of numeric attributes. In *Proceedings of the Tenth Nat. Conference on Artificial Intelligence*, 123-128. MIT Press.
10. Kohavi R., and Sahami M. (1996). Error-based and Entropy-Based Discretization of Continuous Features. In *Proceedings of the Second Int. Conference on Knowledge & Data Mining*, 114-119. AAAI Press.
11. Maas W. (1994). Efficient agnostic PAC-learning with simple hypotheses. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, 67-75. ACM Press.
12. Provost F., and Fawcett T. (1997). Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the Third Int. Conference on Knowledge Discovery and Data Mining*, 43-48, AAAI Press.
13. Van de Merckt T. (1993). Decision Trees in Numerical Attributes Spaces. In *Proceedings of the Thirteenth Int. Joint Conference on Artificial Intelligence*, 1016-1021, Morgan Kaufmann.