A Study of Probability Estimation Techniques for Rule Learning*

Jan-Nikolas Sulzmann and Johannes Fürnkranz

Department of Computer Science, TU Darmstadt
Hochschulstr. 10, D-64289 Darmstadt, Germany
{sulzmann,juffi}@ke.informatik.tu-darmstadt.de

Abstract. Rule learning is known for its descriptive and therefore comprehensible classification models which also yield good class predictions. However, in some application areas, we also need good class probability estimates. For different classification models, such as decision trees, a variety of techniques for obtaining good probability estimates have been proposed and evaluated. However, so far, there has been no systematic empirical study of how these techniques can be adapted to probabilistic rules and how these methods affect the probability-based rankings. In this paper we apply several basic methods for the estimation of class membership probabilities to classification rules. We also study the effect of a shrinkage technique for merging the probability estimates of rules with those of their generalizations.

1 Introduction

The main focus of symbolic learning algorithms such as decision tree and rule learners is to produce a comprehensible explanation for a class variable. Thus, they learn concepts in the form of crisp IF-THEN rules. On the other hand, many practical applications require a finer distinction between examples than is provided by their predicted class labels. For example, one may want to be able to provide a confidence score that estimates the certainty of a prediction, to rank the predictions according to their probability of belonging to a given class, to make a cost-sensitive prediction, or to combine multiple predictions.

All these problems can be solved straight-forwardly if we can predict a probability distribution over all classes instead of a single class value. A straightforward approach to estimate probability distributions for classification rules is to compute the fractions of the covered examples for each class. However, this naïve approach has obvious disadvantages, such as that rules that cover only a few examples may lead to extreme probability estimates. Thus, the probability estimates need to be smoothed.

There has been quite some previous work on probability estimation from decision trees (so-called *probability-estimation trees (PETS)*). A very simple,

* A slightly different version of this paper appeared at the *12th International Conference on Discovery Science*, Porto, 2009.

but quite powerful technique for improving class probability estimates is the use of m-estimates, or their special case, the Laplace-estimates (Cestnik, 1990). Provost and Domingos (2003) showed that unpruned decision trees with Laplace-corrected probability estimates at the leaves produce quite reliable decision tree estimates. Ferri et al. (2003) proposed a recursive computation of the m-estimate, which uses the probability disctribution at level l as the prior probabilities for level $l + 1$. Wang and Zhang (2006) used a general shrinkage approach, which interpolates the estimated class distribution at the leaf nodes with the estimates in interior nodes on the path from the root to the leaf.

An interesting observation is that, contrary to classification, class probability estimation for decision trees typically works better on unpruned trees than on pruned trees. The explanation for this is simply that, as all examples in a leaf receive the same probability estimate, pruned trees provide a much coarser ranking than unpruned trees. Hüllermeier and Vanderlooy (2009) have provided a simple but elegant analysis of this phenomenon, which shows that replacing a leaf with a subtree can only lead to an increase in the area under the ROC curve (AUC), a commonly used measure for the ranking capabilities of an algorithm. Of course, this only holds for the AUC estimate on the training data, but it still may provide a strong indication why unpruned PETs typically also outperform pruned PETs on the test set.

Despite the amount of work on probability estimation for decision trees, there has been hardly any systematic work on probability estimation for rule learning. Despite their obvious similarility, we nevertheless argue that a separate study of probability estimates for rule learning is necessary.

A key difference is that in the case of decision tree learning, probability estimates will not change the prediction for an example, because the predicted class only depends on the probabilities of a single leaf of the tree, and such local probability estimates are typically monotone in the sense that they all maintain the majority class as the class with the maximum probability. In the case of rule learning, on the other hand, each example may be classified by multiple rules, which may possibly predict different classes. As many tie breaking strategies depend on the class probabilities, a local change in the class probability of a single rule may change the global prediction of the rule-based classifier.

Because of these non-local effects, it is not evident that the same methods that work well for decision tree learning will also work well for rule learning. Indeed, as we will see in this paper, our conclusions differ from those that have been drawn from similar experiments in decision tree learning. For example, the above-mentioned argument that unpruned trees will lead to a better (training-set) AUC than pruned trees, does not straight-forwardly carry over to rule learning, because the replacement of a leaf with a subtree is a local operation that only affects the examples that are covered by this leaf. In rule learning, on the other hand, each example may be covered by multiple rules, so that the effect of replacing one rule with multiple, more specific rules is less predictable. Moroever, each example will be covered by some leaf in a decision tree, whereas

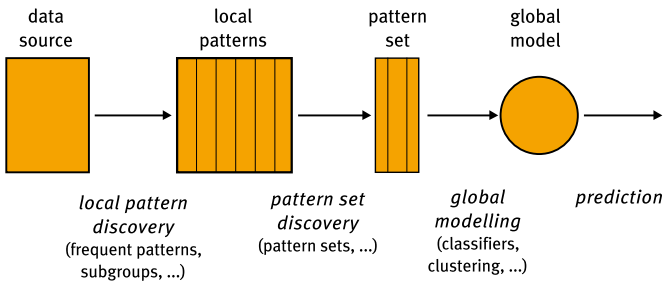

Fig. 1. The LeGo framework (Knobbe et al., 2008)

each rule learner needs to induce a separate default rule that covers examples that are covered by no other rule.

In most cases the probabilistic rule learning task can be divided into three more or less separable phases: the local pattern discovery, which generates a number of candidate rules (local patterns), the pattern set discovery, which selects a rule set (pattern set) from the candidate rules and the global modeling, which generates a global model according to the probability estimates of the rules in the rule set. So probabilistic rule learning may be considered as an example for the recently proposed LeGo data mining framework (see Figure 1) for combining local patterns into a global model (Knobbe et al., 2008).

The rest of the paper is organized as follows: In section 2 we briefly describe the basics of probabilistic rule learning and recapitulate the estimation techniques used for rule probabilities. In section 3 we explain our two approaches for the generation of a probabilistic rule set and describe how it is used for classification. Our experimental setup and results are analyzed in section 4. In the end we summarize our conclusions in section 5.

2 Rule Learning and Probability Estimation

This section is divided into two parts. The first one describes briefly the properties of conjunctive classification rules and of its extension to a probabilistic rule. In the second part we introduce the probability estimation techniques used in this paper. These techniques can be divided into basic methods, which can be used stand-alone for probability estimation, and the meta technique shrinkage, which can be combined with any of the techniques for probability estimation.

2.1 Probabilistic Rule Learning

In classification rule mining one searches for a set of rules that describes the data as accurately as possible. As there are many different generation approaches and

types of generated classification rules, we do not go into detail and restrict ourselves to conjunctive rules. The *premise* of these rules consists of a conjunction of number of conditions, and in our case, the *conclusion* of the rule is a single class value. So a conjunctive classification rule r has basically the following form:

$$condition_1 \wedge \cdots \wedge condition_{|r|} \implies class \tag{1}$$

The size of a rule $|r|$ is the number of its conditions. Each of these conditions consists of an attribute, an attribute value belonging to its domain and a comparison determined by the attribute type. For our purpose, we consider only nominal and numerical attributes. For nominal attributes, this comparison is a test of equality, whereas in the case of numerical attributes, the test is either less (or equal) or greater (or equal). If all conditions are met by an instance, the instance is covered by the rule ($r \supseteq x$) and the class value of the rule is predicted for the instance. Consequently, the rule is called a *covering rule* for this instance.

This in mind, we can define some statistical values of a data set which are needed for later definitions. A data set consists of $|C|$ classes and n instances from which n^c belong to the class c respectively ($n = \sum_{c=1}^{|C|} n^c$). A rule r covers n_r instances which are distributed over the classes, so that n_r^c instances belong to class c ($n_r = \sum_{c=1}^{|C|} n_r^c$).

A probabilistic rule is an extension of a classification rule, which does not only predict a single class value, but a set of *class probabilities*, which form a probability distribution over the classes. This probability distribution estimates all probabilities that a covered instance belongs to any of the class in the data set, so we get one class probability per class. The example is then classified with the most probable class. The probability that an instance x covered by rule r belongs to c can be viewed as a conditional probability $\Pr(c|r \supseteq x)$.

In the next section, we discuss some approaches for estimating these class probabilities.

2.2 Basic Probability Estimation

In this subsection we will review three basic methods for probability estimation. Subsequently, in section 2.3, we will describe a technique known as shrinkage, which is known from various application areas, and show how this technique can be adapted to probabilistic rule learning.

All of the three basic methods we employed, calculate the relation between the number of instances covered by the rule n_r and the number of instances covered by the rule but also belong to a specific class n_r^c. The differences between the methods are the minor modifications of the calculation of this relation.

The simplest approach to rule probability estimation directly estimates a class probability distribution of a rule with the fraction of examples that belong to each class.

$$\Pr_{\text{naïve}}(c|r \supseteq x) = \frac{n_r^c}{n_r} \tag{2}$$

This naïve approach has several well-known disadvantages, most notably that rules with a low coverage may be lead to extreme probability values. For this reason, Cestnik (1990) suggested the use of the Laplace- and m-estimates.

The Laplace estimate modifies the above-mentioned relation by adding one additional instance to the counts n_r^c for each class c. Hence the number of covered instances n_r^c is increased by the number of classes $|C|$.

$$\Pr_{\text{Laplace}}(c|r \supseteq x) = \frac{n_r^c + 1}{n_r + |C|} \tag{3}$$

It may be viewed as a trade-off between $\Pr_{\text{naïve}}(c|r \supseteq x)$ and an *a priori* probability of $\Pr(c) = 1/|C|$ for each class. Thus, it implicitly assumes a uniform class distribution.

The m-estimate generalizes this idea by making the dependency on the prior class distribution explicit, and introducing a parameter m, which allows to trade off the influence of the *a priori* probability and $\Pr_{\text{naïve}}$.

$$\Pr_m(c|r \supseteq x) = \frac{n_r^c + m \cdot \Pr(c)}{n_r + m} \tag{4}$$

The m-parameter may be interpreted as a number of examples that are distributed according to the prior probability, which are added to the class frequencies n_r^c. The prior probability is typically estimated from the data using $\Pr(c) = n^c/n$ (but one could, e.g., also use the above-mentioned Laplace-correction if the class distribution is very skewed). Obviously, the Laplace-estimate is a special case of the m-estimate with $m = |C|$ and $\Pr(c) = 1/|C|$.

2.3 Shrinkage

Shrinkage is a general framework for smoothing probabilities, which has been successfully applied in various research areas.[1] Its key idea is to "shrink" probability estimates towards the estimates of its generalized rules r_k, which cover more examples. This is quite similar to the idea of the Laplace- and m-estimates, with two main differences: First, the shrinkage happens not only with respect to the prior probability (which would correspond to a rule covering all examples) but interpolates between several different generalizations, and second the weights for the trade-off are not specified *a priori* (as with the m-parameter in the m-estimate) but estimated from the data.

In general, shrinkage estimates the probability $\Pr(c|r \supseteq x)$ as follows:

$$\Pr_{Shrink}(c|r \supseteq x) = \sum_{k=0}^{|r|} w_c^k \Pr(c|r_k) \tag{5}$$

where w_c^k are weights that interpolate between the probability estimates of the generalized rules r_k. In our implementation, we use only generalizations of a rule

[1] Shrinkage is, e.g., regularly used in statistical language processing (Chen and Goodman, 1998; Manning and Schütze, 1999)

that can be obtained by deleting a final sequence of conditions. Thus, for a rule with length $|r|$, we obtain $|r|+1$ generalizations r_k, where r_0 is the rule covering all examples, and $r_{|r|} = r$.

The weights w_c^k can be estimated in various ways. We employ a shrinkage method proposed by Wang and Zhang (2006) which is intended for decision tree learning but can be straight-forwardly adapted to rule learning. The authors propose to estimate the weights w_c^k with an iterative procedure which averages the probabilities obtained by removing training examples covered by this rule. In effect, we obtain two probabilities per rule generalization and class: the removal of an example of class c leads to a decreased probability $\Pr_-(c|r_k \supseteq x)$, whereas the removal of an example of a different class results in an increased probability $\Pr_+(c|r_k \supseteq x)$. Weighting these probabilities with the relative occurrence of training examples belonging to this class we obtain a smoothed probability

$$\Pr_{Smoothed}(c|r_k \supseteq x) = \frac{n_r^c}{n_r} \cdot \Pr_-(c|r_k \supseteq x) + \frac{n_r - n_r^c}{n_r} \cdot \Pr_+(c|r_k \supseteq x) \qquad (6)$$

Using these smoothed probabilities, this shrinkage method computes the weights of these nodes in linear time (linear in the number of covered instances) by normalizing the smoothed probabilities separately for each class.

$$w_c^k = \frac{\Pr_{Smoothed}(c|r_k \supseteq x)}{\sum_{i=0}^{|r|} \Pr_{Smoothed}(c|r_i \supseteq x)} \qquad (7)$$

Multiplying the weights with their corresponding probability we obtain "shrinked" class probabilities for the instance.

Note that all instances which are classified by the same rule receive the same probability distribution. Therefore the probability distribution of each rule can be calculated in advance.

3 Rule Learning Algorithm

For the rule generation we employed the the rule learner Ripper (Cohen, 1995), arguably one of the most accurate rule learning algorithms today. We used Ripper both in ordered and in unordered mode:

Ordered Mode: In ordered mode, Ripper learns rules for each class, where the classes are ordered according to ascending class frequencies. For learning the rules of class c_i, examples of all classes c_j with $j > i$ are used as negative examples. No rules are learned for the last and most frequent class, but a rule that implies this class is added as the default rule. At classification time, these rules are meant to be used as a decision list, i.e., the first rule that fires is used for prediction.

Unordered Mode: In unordered mode, Ripper uses a one-against-all strategy for learning a rule set, i.e., one set of rules is learned for each class c_i, using all examples of classes $c_j, j \neq i$ as negative examples. At prediction time, all

rules that cover an example are considered and the rule with the maximum probability estimate is used for classifying the example. If no rule covers the example, it classified by the default rule predicting the majority class.

We used JRip, the Weka (Witten and Frank, 2005) implementation of Ripper. Contrary to William Cohen's original implementation, this re-implementation does not support the unordered mode, so we had to add a re-implementation of that mode.[2] We also added a few other minor modifications which were needed for the probability estimation, e.g. the collection of statistical counts of the sub rules.

In addition, Ripper (and JRip) can turn the incremental reduced error pruning technique (Fürnkranz and Widmer, 1994; Fürnkranz, 1997) on and off. Note, however, that with turned off pruning, Ripper still performs pre-pruning using a minimum description length heuristic (Cohen, 1995). We use Ripper with and without pruning and in ordered and unordered mode to generate four set of rules. For each rule set, we employ several different class probability estimation techniques.

In the test phase, all covering rules are selected for a given test instance. Using this reduced rule set we determine the most probable rule. For this purpose we select the most probable class of each rule and use this class value as the prediction for the given test instance and the class probability for comparison. Ties are solved by predicting the least represented class. If no covering rules exist the class probability distribution of the default rule is used.

4 Experimental Setup

We performed our experiments within the WEKA framework (Witten and Frank, 2005). We tried each of the four configuration of Ripper (unordered/ordered and pruning/no pruning) with 5 different probability estimation techniques, Naïve (labeled as Precision), Laplace, and m-estimate with $m \in \{2, 5, 10\}$, both used as a stand-alone probability estimate (abbreviated with B) or in combination with shrinkage (abbreviated with S). As a baseline, we also included the performance of pruned or unpruned standard JRip accordingly. Additionally our unordered implementation of JRip using Laplace stand-alone for the probability estimation is comparable to the unordered version of Ripper which is not implemented in JRip.

We evaluated these methods on 33 data sets of the UCI repository (Asuncion and Newman, 2007) which differ in the number of attributes (and their categories), classes and training instances. As a performance measure, we used the weighted area under the ROC curve (AUC), as used for probabilistic decision trees by Provost and Domingos (2003). Its key idea is to extend the binary

[2] Weka supports a general one-against-all procedure that can also be combined with JRip, but we could not use this because it did not allow us to directly access the rule probabilities.

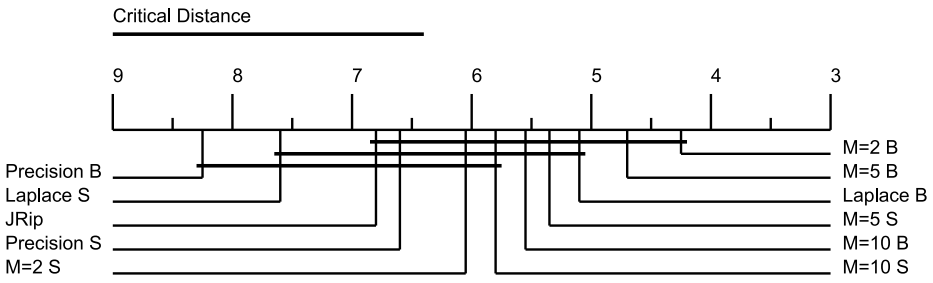

Fig. 2. CD chart for ordered rule sets without pruning

AUC to the multi-class case by computing a weighted average the AUCs of the one-against-all problems N_c, where each class c is paired with all other classes:

$$AUC(N) = \sum_{c \in C} \frac{n_c}{|N|} AUC(N_c) \tag{8}$$

For the evaluation of the results we used the Friedman test with a post-hoc Nemenyi test as proposed in (Demsar, 2006). The significance level was set to 5% for both tests. We only discuss summarized results here, detailed results can be found in the appendix.

4.1 Ordered Rulesets

In the first two test series, we investigated the ordered approach using the standard JRip approach for the rule generation, both with and without pruning. The basic probability methods were used standalone (B) or in combination with shrinkage (S).

The Friedman test showed that in both test series, the employed combinations of probability estimation techniques showed significant differences. Considering the CD chart of the first test series (Figure 2), one can identify three groups of equivalent techniques. Notable is that the two best techniques, the m-Estimate used stand-alone with $m = 2$ and $m = 5$ respectively, belong only to the best group. So they are the only methods that differ significantly from the worst methods that belong to the second and third group and can be therefore considered optimal choices for this scenario. On the other hand, the naïve approach seems to be a bad choice as both techniques employing it rank in the lower half. However our benchmark JRip is positioned in the lower third, which means that the probability estimation techniques clearly improve over the default decision list approach implemented in JRip.

Comparing the stand-alone techniques with those employing shrinkage one can see that shrinkage is outperformed by their stand-alone counterparts. Only precision is an exception as shrinkage yields increased performance in this case. In the end shrinkage is not a good choice for this scenario.

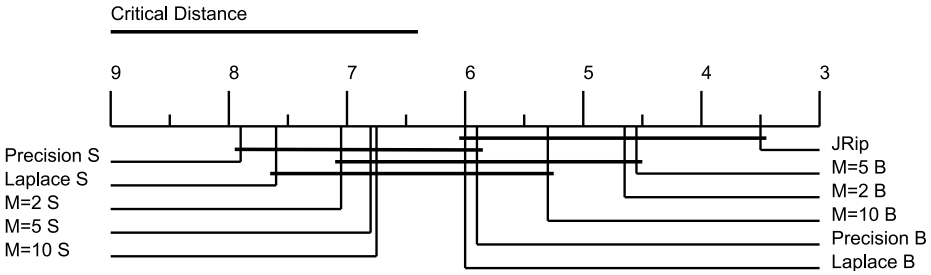

Fig. 3. CD chart for ordered rule sets with pruning

The CD-chart for ordered rule sets with pruning (Figure 3) features four groups of equivalent techniques. Notable are the best and the worst group which overlap only in two techniques, Laplace and Precision used stand-alone. The first group consists of all stand-alone methods and JRip which dominates the group strongly covering no shrinkage method. The last group consists of all shrinkage methods and the overlapping methods Laplace and Precision used stand-alone. As all stand-alone methods rank before the shrinkage methods one can conclude that stand-alone methods outperform the shrinkage methods in this scenario, too. Though performing best in this scenario JRip is indistinguishable from the stand-alone methods.

4.2 Unordered Rule Sets

Test series three and four used the unordered approach employing the modified JRip which generates rules for each class. Analogous to the previous test series the basic methods are used as stand-alone methods or in combination with shrinkage (left and right column respectively). Test series three used no pruning while test series four did so. The results of the Friedman test showed that the techniques of test series three and test series four differ significantly.

Regarding the CD chart of test series three (Figure 4), we can identify four groups of equivalent methods. The first group consists of all stand-alone tech-

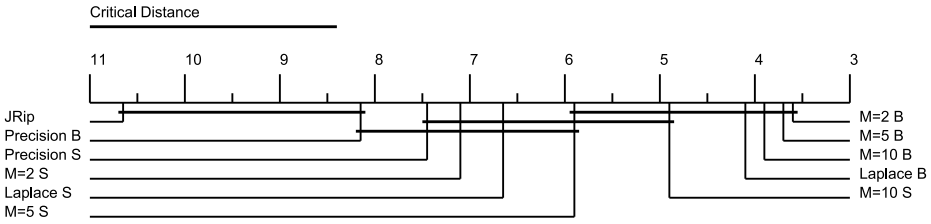

Fig. 4. CD chart for unordered rule sets without pruning

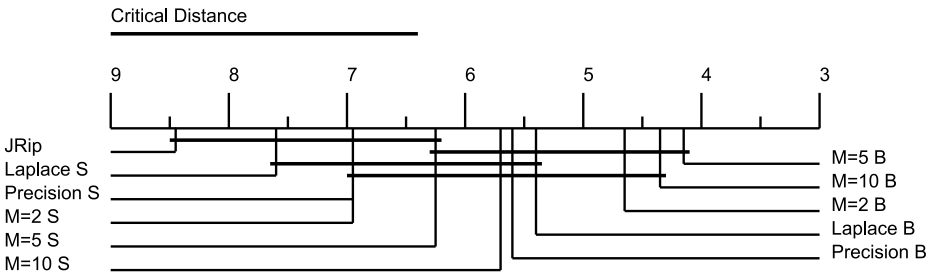

Fig. 5. CD chart for unordered rule sets with pruning

niques, except for Precision, and the m-estimates techniques combined with shrinkage and $m = 5$ and $m = 10$, respectively. Whereas the stand-alone methods dominate this group, $m = 2$ being the best representative, and also belong to this group. Apparently these methods are the best choices for this scenario. The second and third consist mostly of techniques employing shrinkage and overlap with the worst group in only one technique. However our benchmark JRip belongs to the worst group being the worst choice of this scenario. Additionally the shrinkage methods are outperformed by their stand-alone counterparts.

The CD chart of test series four (Figure 5) shows similar results. Again four groups of equivalent techniques groups can be identified. The first group consists of all stand-alone methods and the m-estimates using shrinkage and $m = 5$ and $m = 10$ respectively. This group is dominated by the m-estimates used stand-alone with $m = 2$, $m = 5$ or $m = 10$. The shrinkage methods are distributed over the other groups, again occupying the lower half of the ranking. Our benchmark JRip is the worst method of this scenario.

4.3 Unpruned vs. Pruned Rule Sets

Rule pruning had mixed results, which are briefly summarized in Table 1. On the one hand, it improved the results of the unordered approach, on the other hand it worsened the results of the ordered approach. In any case, in our experiments, contrary to previous results on PETs, rule pruning was not always a bad choice. The explanation for this result is that in rule learning, contrary to decision tree learning, new examples are not necessarily covered by one of the learned rules. The more specific rules become, the higher is the chance that new examples are not covered by any of the rules and have to be classified with a default rule. As these examples will all get the same default probability, this is a bad strategy for probability estimation. Note, however, that JRip without pruning, as used in our experiments, still performs an MDL-based form of pre-pruning. We have not yet tested a rule learner that performs no pruning at all, but, because of the above deliberations, we do not expect that this would change the results with respect to pruning.

Table 1. Unpruned vs. pruned rule sets: Win/Loss for ordered (top) and unordered (bottom) rule sets

	Jrip	Precision		Laplace		M 2		M 5		M 10	
Win	26	23	19	20	19	18	20	19	20	19	20
Loss	7	10	14	13	14	15	13	14	13	14	13
Win	26	21	9	8	8	8	8	8	8	8	6
Loss	7	12	24	25	25	25	25	25	25	25	27

5 Conclusions

The most important result of our study is that probability estimation is clearly an important part of a good rule learning algorithm. The probabilities of rules induced by JRip can be improved considerably by simple estimation techniques. In unordered mode, where one rule is generated for each class, JRip is outperformed in every scenario. On the other hand, in the ordered setting, which essentially learns decision lists by learning subsequent rules in the context of previous rules, the results were less convincing, giving a clear indication that the unordered rule induction mode should be preferred when a probabilistic classfication is desirable.

Amongst the tested probability estimation techniques, the m-estimate typically outperformed the other methods. Among the tested values, $m = 5$ seemed to yield the best overall results, but the superiority of the m-estimate was not sensitive to the choice of this parameter. The employed shrinkage method did in general not improve the simple estimation techniques. It remains to be seen whether alternative ways of setting the weights could yield superior results. Rule pruning had mixed results, so contrary to PETs pruning is not always a bad choice.

In (Sulzmann and Fürnkranz, 2008) we surveyed and evaluated several options for selecting a subset of class association rules and for combining their predictions into a global rule model. Our next step will be to investigate how the generation algorithm for classification rules deployed in this paper can be modified for the generation of probabilistic rules and how the selecting and combining strategies perform on probabilistic rules. Thus, we hope to obtain a solid, well-founded procedure for obtaining probabilistic classifiers from local patterns.

Acknowledgements

This research was supported by the *German Science Foundation (DFG)* under grant FU 580/2.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`.

Bojan Cestnik. Estimating probabilities: A crucial task in Machine Learning. In L. Aiello, editor, *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90)*, pages 147–150, Stockholm, Sweden, 1990. Pitman.

Stanley F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, 1998.

William W. Cohen. Fast effective rule induction. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ML-95)*, pages 115–123, Lake Tahoe, CA, 1995. Morgan Kaufmann.

Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

César Ferri, Peter A. Flach, and José Hernández-Orallo. Improving the AUC of probabilistic estimation trees. In *Proceedings of the 14th European Conference on Machine Learning*, pages 121–132, Cavtat-Dubrovnik, Croatia, 2003.

Johannes Fürnkranz. Pruning algorithms for rule learning. *Machine Learning*, 27(2):139–171, 1997. URL `http://www.ke.informatik.tu-darmstadt.de/~juffi/publications/mlj97.pdf`.

Johannes Fürnkranz and Peter A. Flach. Roc 'n' rule learning-towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.

Johannes Fürnkranz and Gerhard Widmer. Incremental Reduced Error Pruning. In W. Cohen and H. Hirsh, editors, *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 70–77, New Brunswick, NJ, 1994. Morgan Kaufmann. URL `http://www.ke.informatik.tu-darmstadt.de/~juffi/publications/ml-94.ps.g%z`.

David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2): 171–186, 2001.

Eyke Hüllermeier and Stijn Vanderlooy. Why fuzzy decision trees are good rankers. *IEEE Transactions on Fuzzy Systems*, to appear, 2009.

Arno Knobbe, Bruno Crémilleux, Johannes Fürnkranz, and Martin Scholz. From local patterns to global models: The LeGo approach to data mining. In J. Fürnkranz and A. Knobbe (eds.) *From Local Patterns to Global Models: Proceedings of the ECML/PKDD-08 Workshop (LeGo-08)*, pp. 1–16, Antwerp, Belgium, 2008.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

Foster J. Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003.

Jan-Nikolas Sulzmann and, Johannes Fürnkranz. An empirical comparison of techniques for selecting and combining local patterns into a global model. Technical Report TUD-KE-2008-03, Knowledge Engineering Group, Technische Universität Darmstadt, Hochschulstrasse 10, D-64289 Darmstadt, Germany, 2008.

Bin Wang and Harry Zhang. Improving the ranking performance of decision trees. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proceedings of the 17th European Conference on Machine Learning (ECML-06)*, pages 461–472, Berlin, Germany, 2006. Springer-Verlag.

Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition.* Morgan Kaufmann, San Francisco, 2005.

A Detailed experimental results (tables)

Table 2. Weighted AUC results with rules from ordered, unpruned JRip.

Name	Jrip	Precision		Laplace		M 2		M 5		M 10	
		B	S	B	S	B	S	B	S	B	S
Anneal	.983	.970	.971	.970	.970	.970	.970	.971	.971	.970	.970
Anneal.orig	.921	.917	.920	.920	.919	.920	.919	.920	.920	.919	.919
Audiology	.863	.845	.843	.832	.836	.840	.844	.839	.841	.831	.832
Autos	.904	.907	.901	.900	.891	.907	.902	.904	.902	.903	.898
Balance-scale	.823	.801	.812	.821	.812	.820	.811	.821	.812	.821	.815
Breast-cancer	.591	.577	.581	.578	.580	.578	.580	.578	.579	.577	.579
Breast-w	.928	.930	.929	.935	.930	.935	.931	.935	.932	.935	.933
Colic	.736	.739	.741	.747	.746	.748	.746	.748	.745	.748	.746
Credit-a	.842	.849	.857	.861	.859	.861	.859	.861	.862	.861	.864
Credit-g	.585	.587	.587	.587	.587	.587	.587	.587	.587	.587	.587
Diabetes	.642	.654	.656	.655	.656	.655	.656	.655	.656	.655	.655
Glass	.806	.803	.795	.790	.787	.794	.797	.793	.799	.792	.795
Heart-c	.762	.765	.775	.796	.777	.796	.777	.796	.780	.796	.789
Heart-h	.728	.737	.755	.758	.757	.758	.755	.758	.757	.758	.757
Heart-statlog	.763	.759	.781	.806	.782	.806	.783	.806	.790	.806	.791
Hepatitis	.679	.661	.661	.660	.663	.660	.665	.660	.663	.660	.663
Hypothyroid	.971	.973	.974	.974	.974	.974	.974	.974	.974	.973	.974
Ionosphere	.884	.885	.897	.903	.900	.903	.899	.903	.900	.903	.902
Iris	.957	.889	.876	.889	.878	.889	.878	.889	.878	.889	.878
Kr-vs-kp	.993	.994	.994	.995	.994	.995	.994	.995	.994	.995	.994
Labor	.812	.800	.810	.794	.810	.793	.806	.793	.795	.793	.783
Lymph	.750	.739	.748	.748	.745	.746	.748	.744	.746	.749	.746
Primary-tumor	.649	.636	.652	.615	.638	.645	.656	.641	.653	.642	.662
Segment	.983	.964	.944	.967	.943	.966	.944	.967	.943	.966	.943
Sick	.922	.928	.929	.929	.929	.929	.929	.929	.929	.929	.929
Sonar	.774	.771	.779	.784	.778	.783	.778	.783	.779	.783	.781
Soybean	.962	.971	.972	.966	.971	.973	.972	.967	.973	.967	.971
Splice	.938	.934	.938	.943	.938	.943	.938	.943	.938	.943	.939
Vehicle	.772	.799	.811	.811	.816	.812	.813	.811	.816	.812	.819
Vote	.952	.954	.950	.955	.949	.955	.949	.955	.952	.953	.956
Vowel	.884	.906	.909	.909	.906	.909	.910	.911	.910	.910	.907
Waveform	.847	.850	.853	.872	.854	.872	.854	.873	.855	.873	.858
Zoo	.916	.899	.916	.902	.897	.908	.900	.907	.895	.899	.890
Average	.834	.830	.834	.836	.832	.837	.834	.837	.834	.836	.834
Average Rank	6.79	8.24	6.62	5.11	7.62	4.26	6.03	4.68	5.33	5.53	5.79

Table 3. Weighted AUC results with rules from ordered, pruned JRip.

Name	Jrip	Precision		Laplace		M 2		M 5		M 10	
		B	S	B	S	B	S	B	S	B	S
Anneal	.984	.981	.980	.981	.981	.981	.980	.981	.980	.980	.980
Anneal.orig	.942	.938	.937	.936	.936	.937	.936	.936	.937	.935	.936
Audiology	.907	.865	.854	.810	.776	.852	.840	.839	.826	.834	.801
Autos	.850	.833	.836	.821	.829	.829	.830	.823	.830	.821	.819
Balance-scale	.852	.812	.810	.815	.810	.815	.810	.816	.811	.816	.811
Breast-cancer	.598	.596	.597	.596	.597	.596	.597	.598	.599	.598	.602
Breast-w	.973	.965	.956	.965	.956	.964	.956	.964	.957	.961	.957
Colic	.823	.801	.808	.804	.815	.809	.815	.813	.815	.816	.816
Credit-a	.874	.872	.874	.873	.874	.874	.874	.874	.873	.875	.874
Credit-g	.593	.613	.612	.613	.612	.613	.612	.613	.612	.613	.612
Diabetes	.739	.734	.736	.734	.736	.734	.736	.734	.736	.734	.736
Glass	.803	.814	.810	.822	.825	.820	.818	.820	.817	.820	.812
Heart-c	.831	.837	.818	.843	.818	.842	.818	.845	.823	.847	.825
Heart-h	.758	.739	.742	.740	.740	.740	.742	.741	.742	.742	.741
Heart-statlog	.781	.792	.776	.790	.776	.790	.776	.791	.775	.790	.773
Hepatitis	.664	.600	.596	.600	.596	.599	.596	.599	.595	.597	.586
Hypothyroid	.988	.990	.990	.990	.990	.990	.990	.990	.990	.990	.990
Ionosphere	.900	.904	.909	.907	.909	.908	.909	.910	.910	.910	.909
Iris	.974	.888	.889	.890	.891	.890	.891	.890	.891	.890	.891
Kr-vs-kp	.995	.994	.993	.994	.993	.994	.993	.994	.994	.994	.994
Labor	.779	.782	.755	.782	.761	.781	.764	.768	.759	.746	.745
Lymph	.795	.795	.767	.788	.772	.790	.773	.779	.773	.777	.774
Primary-tumor	.642	.626	.624	.622	.627	.630	.622	.627	.622	.629	.628
Segment	.988	.953	.932	.953	.933	.954	.932	.953	.932	.953	.933
Sick	.948	.949	.949	.950	.949	.950	.949	.950	.950	.950	.950
Sonar	.759	.740	.734	.742	.737	.743	.737	.746	.740	.744	.744
Soybean	.981	.980	.970	.968	.965	.978	.970	.971	.967	.969	.966
Splice	.967	.956	.953	.957	.953	.957	.953	.957	.954	.957	.954
Vehicle	.855	.843	.839	.844	.843	.844	.842	.843	.843	.842	.844
Vote	.942	.949	.947	.949	.947	.949	.947	.949	.947	.949	.947
Vowel	.910	.900	.891	.898	.891	.904	.892	.905	.893	.898	.892
Waveform	.887	.880	.862	.880	.863	.880	.862	.881	.863	.881	.863
Zoo	.925	.889	.909	.887	.895	.895	.902	.895	.901	.889	.893
Average	.855	.843	.838	.841	.836	.843	.838	.842	.838	.841	.836
Average Rank	3.52	5.88	7.92	5.98	7.62	4.65	7.06	4.55	6.79	5.29	6.74

Table 4. Weighted AUC results with rules from unordered, unpruned JRip.

Name	Jrip	Precision		Laplace		M 2		M 5		M 10	
		B	S	B	S	B	S	B	S	B	S
Anneal	.983	.992	.989	.992	.991	.994	.989	.994	.989	.994	.989
Anneal.orig	.921	.987	.984	.990	.983	.993	.984	.993	.984	.993	.984
Audiology	.863	.910	.887	.877	.874	.909	.895	.903	.894	.892	.889
Autos	.904	.916	.915	.926	.914	.927	.914	.929	.918	.930	.926
Balance-scale	.823	.874	.865	.908	.873	.908	.866	.909	.871	.908	.882
Breast-cancer	.591	.608	.587	.633	.605	.633	.589	.632	.606	.632	.617
Breast-w	.928	.959	.966	.953	.966	.953	.967	.953	.969	.953	.969
Colic	.736	.835	.840	.855	.851	.855	.849	.855	.849	.859	.849
Credit-a	.842	.890	.909	.913	.911	.913	.911	.913	.914	.913	.917
Credit-g	.585	.695	.717	.716	.716	.716	.716	.716	.716	.716	.718
Diabetes	.642	.760	.778	.783	.780	.783	.779	.783	.781	.783	.783
Glass	.806	.810	.826	.808	.833	.808	.825	.808	.827	.809	.830
Heart-c	.762	.790	.813	.861	.827	.861	.823	.861	.831	.861	.844
Heart-h	.728	.789	.803	.851	.839	.853	.819	.849	.835	.852	.837
Heart-statlog	.763	.788	.811	.845	.805	.841	.805	.841	.820	.841	.829
Hepatitis	.679	.774	.817	.799	.819	.802	.821	.802	.817	.802	.816
Hypothyroid	.971	.991	.994	.994	.993	.994	.994	.994	.993	.994	.993
Ionosphere	.884	.918	.932	.938	.931	.938	.931	.938	.931	.939	.935
Iris	.957	.968	.973	.978	.980	.978	.976	.978	.980	.978	.980
Kr-vs-kp	.993	.998	.997	.999	.997	.999	.997	.999	.997	.999	.997
Labor	.812	.818	.806	.777	.803	.778	.803	.778	.790	.778	.775
Lymph	.750	.843	.852	.891	.857	.887	.848	.881	.852	.884	.878
Primary-tumor	.649	.682	.707	.671	.690	.693	.712	.694	.711	.691	.711
Segment	.983	.991	.989	.997	.990	.997	.989	.997	.990	.997	.990
Sick	.922	.958	.979	.981	.984	.982	.979	.982	.980	.982	.980
Sonar	.774	.823	.826	.841	.826	.841	.826	.841	.828	.841	.836
Soybean	.962	.979	.981	.982	.979	.985	.981	.984	.981	.985	.981
Splice	.938	.964	.968	.974	.968	.974	.968	.974	.969	.974	.970
Vehicle	.772	.851	.879	.888	.881	.888	.879	.888	.881	.888	.884
Vote	.952	.973	.967	.982	.968	.983	.968	.983	.975	.983	.978
Vowel	.884	.917	.919	.922	.920	.922	.921	.922	.920	.922	.920
Waveform	.847	.872	.890	.902	.890	.902	.890	.902	.890	.902	.893
Zoo	.916	.964	.965	.965	.970	.984	.982	.984	.982	.987	.988
Average	.834	.875	.883	.891	.885	.893	.885	.893	.887	.893	.890
Average Rank	10.67	8.15	7.45	4.08	6.65	3.58	7.08	3.68	5.88	3.88	4.91

Table 5. Weighted AUC results with rules from unordered, pruned JRip.

Name	Jrip	Precision		Laplace		M 2		M 5		M 10	
		B	S	B	S	B	S	B	S	B	S
Anneal	.984	.987	.988	.984	.986	.987	.985	.986	.986	.986	.986
Anneal.orig	.942	.990	.983	.985	.980	.989	.983	.988	.982	.984	.982
Audiology	.907	.912	.889	.891	.878	.895	.893	.889	.885	.883	.881
Autos	.850	.889	.882	.891	.889	.894	.888	.892	.889	.891	.889
Balance-scale	.852	.888	.861	.899	.864	.895	.860	.900	.861	.901	.864
Breast-cancer	.598	.562	.555	.557	.555	.557	.555	.557	.555	.560	.558
Breast-w	.973	.962	.972	.963	.973	.963	.973	.963	.973	.961	.974
Colic	.823	.782	.831	.799	.830	.793	.836	.801	.837	.812	.837
Credit-a	.874	.876	.878	.877	.877	.877	.878	.879	.879	.881	.879
Credit-g	.593	.702	.711	.703	.711	.703	.711	.703	.711	.705	.711
Diabetes	.739	.740	.729	.742	.729	.742	.729	.741	.730	.739	.731
Glass	.803	.819	.821	.821	.826	.819	.821	.824	.824	.828	.825
Heart-c	.831	.827	.816	.827	.804	.829	.816	.828	.810	.830	.807
Heart-h	.758	.739	.740	.735	.736	.737	.738	.736	.737	.735	.736
Heart-statlog	.781	.806	.815	.816	.813	.816	.812	.823	.819	.824	.827
Hepatitis	.664	.766	.790	.769	.793	.771	.790	.764	.795	.768	.789
Hypothyroid	.988	.984	.993	.992	.993	.987	.994	.992	.993	.992	.993
Ionosphere	.900	.918	.915	.921	.917	.922	.918	.926	.923	.926	.923
Iris	.974	.975	.969	.975	.969	.975	.969	.975	.970	.975	.973
Kr-vs-kp	.995	.999	.995	.999	.995	.999	.995	.999	.996	.998	.997
Labor	.779	.837	.820	.815	.811	.812	.818	.812	.812	.809	.803
Lymph	.795	.858	.832	.849	.833	.853	.836	.851	.842	.851	.856
Primary-tumor	.642	.703	.701	.679	.694	.709	.704	.710	.706	.708	.707
Segment	.988	.991	.989	.995	.990	.995	.990	.995	.990	.995	.990
Sick	.948	.949	.934	.948	.938	.948	.935	.948	.937	.948	.937
Sonar	.759	.827	.815	.827	.814	.827	.815	.824	.813	.824	.818
Soybean	.981	.989	.981	.988	.981	.990	.981	.989	.981	.989	.981
Splice	.967	.973	.967	.974	.967	.974	.967	.974	.968	.974	.968
Vehicle	.855	.892	.891	.893	.890	.893	.890	.893	.890	.893	.890
Vote	.942	.947	.956	.961	.957	.952	.957	.960	.956	.961	.958
Vowel	.910	.921	.915	.924	.915	.925	.915	.925	.916	.924	.915
Waveform	.887	.897	.877	.899	.878	.898	.877	.899	.878	.900	.880
Zoo	.925	.973	.989	.960	.969	.987	.989	.987	.989	.987	.989
Average	.855	.875	.873	.874	.871	.876	.873	.877	.874	.877	.874
Average Rank	8.45	5.61	6.95	5.38	7.59	4.67	6.95	4.14	6.23	4.33	5.7