Learning to Rank Cases with Classification Rules

Jianping Zhang, Jerzy W. Bala, Ali Hadjarian, Brent Han

The MITRE Corporation, 7515 Colshire Drive,
McLean, Virginia 22102-7508 USA
{ jzhang, ahadjarian, bhan}@mitre.org
jerzy.w.bala@gmail.com

Abstract. An advantage of rule induction over other machine learning algorithms is the comprehensibility of the models, a requirement for many data mining applications. However, many real life machine learning applications involve the ranking of cases and classification rules are not a good representation for this. There have been numerous studies to incorporate ranking capability into decision trees, but not rules. We propose a framework for ranking with rules. The framework extends and substantially improves on the reported methods of using decision trees for ranking. It introduces three types of rule ranking methods: post analysis of rules, hybrid methods, and multiple rule set analysis. We also study the impact of rule learning bias on the ranking performance. An empirical study has been conducted to evaluate some of the above methods. The results have been compared with those reported for a decision tree with ranking capability.

Keywords: Learning to Rank, Preference Learning, Rule Induction.

1 Introduction

Classification accuracy has been used as a major performance metric for machine learning algorithms. However, many real life machine learning applications involve the ranking of cases instead of their classification. Ranking of cases is usually based on some kind of reliability, likelihood or numeric assessment of the quality of each classification (e.g., a probability value of a class membership). In other words, the decision-making process extends the class membership prediction to include an estimate of the reliability for this prediction. For example, in credit application processing, the goal is to rank applicants in terms of their likelihoods of profitability and/or to predict loan defaults. This is significantly different than simply classifying them into qualified or non-qualified groups. Other decision-making applications where case ranking could be of importance include bankruptcy prediction, medical diagnosis, customer targeting for marketing campaigns and customer churn prediction. In addition, the use of rule based models facilitates comprehensibility of the ranking process, which may represent an essential requirement for the decision ranking applications. Ranking of cases is also important for those decision-making applications where it is better not to make a decision without sufficient support than to make a wrong decision (e.g., medical and military applications). Such abstention in decision-making may be a preferable in many situations.

The machine learning community has investigated the incorporation of ranking capability within a decision tree learning paradigm [9] [10] [11] [1]. However, little work has been done in ranking with rules. Although rules are similar to decision trees, there are also some important differences between them when used for ranking. Separate-&-conquer (covering) techniques of

rule learning algorithms may generate rules that overlap, whereas divide-&-conquer techniques of decision trees do not. Rules may not cover some areas of a feature space, but leaf nodes of a decision tree cover the entire area of the feature space (Figure 1). These differences bring both research challenges and opportunities for developing methods of ranking cases with rules. In addition to the above differences, a rule learning algorithm for a two class problem may only learn rules for one class, but a decision tree always includes leaf nodes for both classes. Rule learning algorithms tend to generate fewer rules than the number of leaf nodes of a decision tree.

In this paper, we propose a framework for ranking with rules. Specifically, the framework presented in this paper extends and substantially improves on the reported methods for using decision trees for ranking. It introduces three types of rule ranking methods: post analysis of rules, hybrid methods, and multiple rule sets (rule ensembles and redundant rules).

Methods for combining scores of overlapping rules are proposed and studied. We also investigate the impact of rule learning bias on the ranking performance. An empirical study has been conducted to evaluate some popular post analysis methods, methods for combining scores of overlapping rules, and impact of rule learning bias.

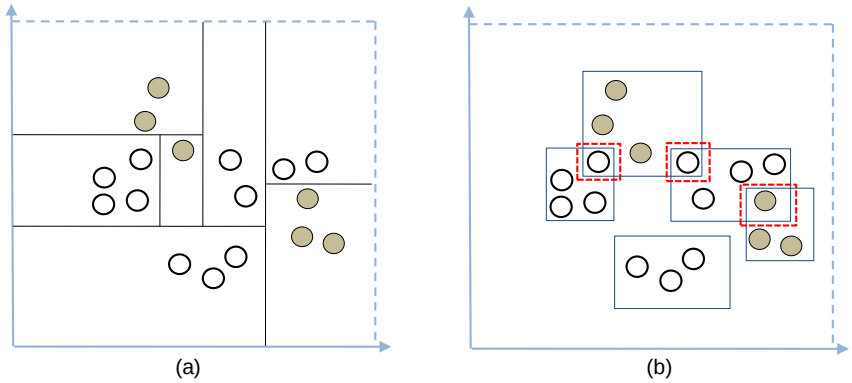

Fig 1. Different modeling paradigms, (a) hyperplanes vs. (b) rule based representations)

The reminder of the paper is organized as follows. Related work is discussed in Section 2. Section 3 describes the proposed framework. Section 4 reports on the experimental results. Finally, Section 5 concludes the paper with future research.

2 Previous Work

The idea of ranking examples using rules is not new. There is related research in expert systems, fuzzy logic, and cognitive science [2]. This section provides a summary of this research as well as the more closely related machine learning work.

2.1 Expert System, Fuzzy Logic, and Cognitive Science

MYCIN [4] is a well known rule-based expert system, in which each rule is assigned a certainty factor (CF) by domain experts. CF of different rules may be combined and/or propagated to

produce the CF of a decision inferred by MYCIN. In PROSPECTOR [12], an expert system to assist geologists working in mineral exploration, rules are assigned probability by human experts and are propagated and combined using Bayesian inference.

Development of fuzzy rules has been studied widely in fuzzy logic. Examples include the work by Wang and Mendel [13], where a general method is developed to generate fuzzy rules from numerical data using linguistic variables. The degree of membership of an example depends on the degree of match of the example to a fuzzy rule.

These attempts have resulted in methods for generating partial matching or fuzzy rules and examples are assigned scores based on how well they match those rules. In these approaches, examples that satisfy all conditions of a rule share the same score. Our proposed method, on the other hand, could assign different scores to different examples even when they satisfy all the conditions of a given rule.

2.2 Rule Ranking in Machine Learning

Previous machine learning research in ranking cases with rules include the study by Zhang and Michalski [14]. They developed a method for generating partial matching and scores are computed based on the degree of match of an example to the rules. Here, once again, examples that satisfy all conditions of a rule share the same score.

The machine learning community has also investigated the incorporation of ranking capability within a decision tree learning paradigm. Most previous work falls within the following three groups of methods:

- Learning Probability Estimation Trees. This group of methods has been reported by Provost and Domingos [11]. To alleviate the problem of uniform probability estimation (a defect that results in the poor probability-based ranking by decision trees), they proposed several techniques to improve the ranking performance of C4.5: turning off pruning and collapsing mechanism to keep the branches that contribute most to the quality of ranking and using Laplace correction at leave nodes for smoothing. They also pointed out that bagging, an ensemble method, could greatly improve decision trees in terms of probability-based ranking.

- Tree Branching. Ferri et al. [7] introduced a method, called m-Branch, that generates probability estimates at leaves in a recursive manner so that on each path, the probability estimates on a parent node are propagated downwards to all of its children. Ling and Yan [10] report on the technique of averaging probability estimates from all the leaves of a tree. The contribution of each leaf is determined by the number of unequal parent attribute values (parent attributes of a leaf are defined as the attributes on the path from the leaf to the root) that the leaf has, compared with the unlabeled cases.

- Hybrid Decision Trees. Kohavi [9] proposed a Naïve Bayes Tree (NBTree) that is a hybrid of decision tree and Naïve Bayes classifiers, where a Naïve Bayes classifier is deployed at each leaf to produce classification and probability estimation.

3 A Framework for Ranking with Classification Rules

A rule based classifier is typically defined as a disjunctive normal form of conditional rules that defines a mapping function from a set of *m* arguments or attributes (which can be either

nominal or numeric) to a single nominal value, known as the class. Let us represent by *D* the set of *d* classes, usually simply referred by natural numbers 0, 1, 2, ... *d*-1, and by *E* the set of examples. A classifier is a function *f*: *E* -> *D*. In a simplified scenario, a classifier with ranking capability computes a number or score for every example $e \in$ E and for every class $i \in D$.

A score may be interpreted differently depending on the application. In statistics, a score ranges from 0 to 1 and means the probability of the example belonging to a given class. In expert systems, a score may be interpreted as a certainty factor. In fuzzy logic, a score represents the degree of membership of the class. In cognitive science, a score could be defined as the typicality of the example being a member of the class. In other applications, it is just a score for ranking cases.

The framework for ranking cases with classification rules presented in this paper consists of a set of different group of methods that can be used independently or jointly. Figure 2 depicts the taxonomy of methods. The framework extends and substantially improves on the reported methods of using decision trees for ranking. The framework introduces the following three groups of rule ranking computation methods:

- Post analysis of rules
- Hybrid methods
- Rule ensembles and redundant rules

In addition to the three groups above, we introduce methods for combining scores for overlapping rules. We also discuss the potential impact of variations of rule induction algorithms.

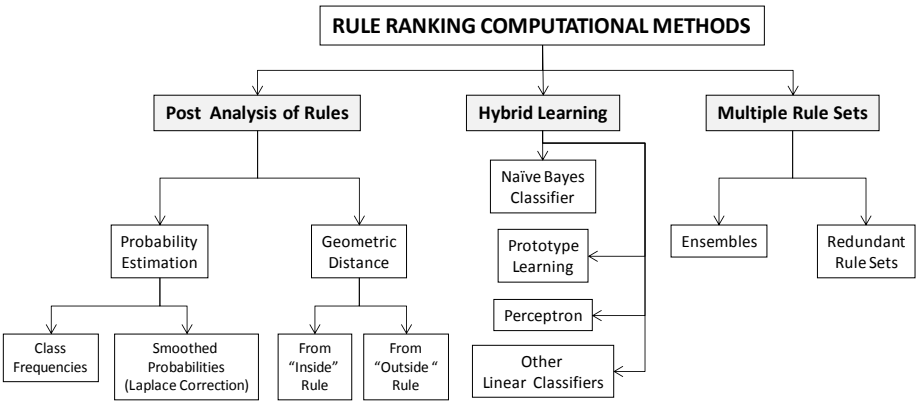

Fig 2. Rule ranking computational methods

3.1 Post Analysis of Rules

In this group of methods, ranking scores are computed using rules generated by some rule induction algorithm plus additional information such as the number of positive and negative examples covered by each rule and/or ranges of attribute values. There are two subgroups of methods under this group, probability estimation and geometric methods.

3.1.1 Probability Estimation

With probability estimation, the score of the example covered by a rule is computed as the ratio of the number of positive examples to the total number of examples covered by that rule. This approach has been explored in Probability Estimation Trees.

In this simple probability estimation method, scores assigned by two different rules are identical as long as the above mentioned ratio is the same, no matter how many positive examples are actually covered by each rule. For example, all rules covering no negative examples result in the same score, namely one. A simple method for overcoming this problem is the Laplace correction, in which the score is computed using

$$\frac{k+1}{n+C}$$

where k and n are the numbers of positive examples and the total number of examples covered, respectively, and C is the number of classes. The score is the same for each example covered by the same rule. This becomes a serious problem when the number of rules generated by a rule induction algorithm is small.

The probability estimation method, even with Laplace correction, ignores the absolute number of examples covered by the rule. A rule covering 9 positive examples and 10 negative examples receives the same score as a rule covering 90 positive examples and 100 negative examples. In real applications, however, even with similar precisions, rules covering more examples are preferred. An alternative option is to use F-measure for scoring:

$$F-measure(r)=\frac{\beta^2+1}{\frac{\beta^2}{recall(r)}+\frac{1}{precision(r)}}$$

where β is a parameter for assigning relative weights to recall and precision. When β is set to 1, recall and precision are weighted equally. F-measure favors recall with $\beta > 1$ and favors precision with $\beta < 1$. Namely, an F-measure with a large β value favors more general and less precise rules, while one with a small β value favors more specific and precise rules. When $\beta = 0$, the F-measure score is the same as probability estimation.

3.1.2 Geometric Methods

As indicated above, the probability estimation method assigns the same score to all examples covered by the same rule. When the number of rules generated is small, this causes a problem for applications that need a fine grained ranking. For example, in one of our data mining applications, only 0.1% of the top ranked cases are selected for further investigations. If all rules cover more than 0.1% of the examples, we have no way to accurately select 0.1% of the top ranked cases. Geometric methods can help generate such fine grained rankings.

Geometric methods have been used in decision trees [1] and assume that classifications/rankings of the examples near the rule boundary are less certain. In ranking with rules, there are two types of geometric methods, one for examples covered by a rule and one for examples that are not covered by any rules. The latter is also called partial matching. For an example covered by a rule, we can measure the distance between the example and the rule boundary or the center of the rule. The closer the example is to the boundary (or further away from the center), the smaller its score gets. The distance may also be weighted by the estimated probability of the rule. The geometric method for covered examples works for numeric attributes.

Partial matching also computes the distance of an uncovered example to the boundary of a rule, but from outside of the rule. The closer to the boundary, the larger the score is. Again, the

distance could be weighted by the estimated probability of the rule. A different view of partial matching is that partial matching differentiates from strict matching, which tests whether an example strictly satisfies a rule (satisfies all conditions). Partial matching computes a degree of match between an example and a rule. The degree of match can vary in the range from 0 (matches no condition) to 1.0 (matches all conditions). Partial matching works for both numeric and nominal attributes.

3.2 Hybrid Methods

Hybrid methods integrate rule induction with other learning techniques that have ranking capabilities. These latter techniques include the Perceptron algorithm, Naïve Bayes, instance-based learning, and prototype-based learning. For example, within each rule, a Perceptron may be learned. Figure 3 shows a rule with a linear classifier in a two dimensional space. The rectangle represents the rule and the line inside the rectangle represents the linear classifier. The white circles represent positive examples, while the gray ones represent negative examples. The linear classifier is used to assign a score to each of the examples covered by the rule.

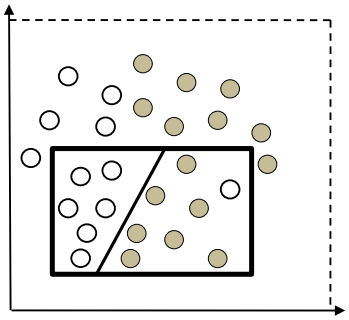

Fig 3. A rule with a linear classifier

Rules and linear models or Naïve Bayes models may be learned together to optimize the performance of the hybrid method. Alternatively, rules may be learned first, and then other models are learned for each of the learned rules. As discussed in Section 2, many works have been done in building hybrid decision trees, e.g., Perceptron Tree and NBTree.

3.3 Rule Ensembles and Redundant Rules

Previous studies have shown that ensemble techniques can significantly improve the ranking performance of decision trees [3] [11]. In ensemble learning, multiple classifiers are learned, using approaches such as Bagging and Boosting. Typically, in such ensembles, a majority vote technique is used to determine the final classification of an example. Similarly, the average score can be used for assignment of a final score to each example. Although little work has been done in building ensembles of rules (e.g., [8]), we believe such an ensemble can yield improved ranking performance.

Previous studies have also shown how redundant rules could improve classification performance. Since redundant rules allow for finer grained rankings, it is worthwhile to design a rule induction algorithm for generating redundant rules for ranking.

3.4 Computing Scores with Multiple Rules

A rule induction algorithm usually generates a set of overlapping rules. In this section, we propose and discuss three simple methods, *Max*, *Average*, and *Probabilistic Sum*, for combining scores of an example covered by more than one rule. The *Max* approach simply takes the largest score of all the rules that cover the example. Given an example *e* and a set of *l* rules RS = {R_1, …, R_l,}, the combined score of *e* using *Max* is computed as follows:

$$score(e, RS) = \max_{i=1}^{l} \{score(e, R_i)\},$$

where *score(e, R_i)* is the score of *e* assigned by Rule R_i. The combined score of *e* using *Average* is computed as follows.

$$score(e, RS) = \frac{\sum_{i=1}^{l} score(e, R_i)}{l}.$$

For rules that do not cover *e*, *score(e, R_i)* = 0 unless partial matching is applied. For the *Probabilistic Sum* method, the formula can be defined recursively as follows:

$$score(e, \{R_1\}) = score(e, R_1)$$

$$score(e, \{R_1, R_2\}) = score(e, R_1) + score(e, R_2) - score(e, R_1) \times score(e, R_2)$$

$$score(e, \{R_1, ..., R_n\}) = score(e, \{R_1, ..., R_{n-1}\}) + score(e, R_n) - score(e, \{R_1, ..., R_{n-1}\}) \times score(e, R_n)$$

Both *Average* and *Probabilistic Sum* generate a finer grained ranking than *Max*. These three methods may also be used to combine scores of ensembles of rules.

3.5 Impact of Rule Induction Algorithm

Most rule induction algorithms have been designed for maximizing the classification accuracy. Recently, some learning algorithms have been proposed that optimize the AUC (Area Under the Curve) of a ROC (Receiver Operating Characteristics) curve [6]. The design of a rule induction algorithm for optimizing AUC could be an interesting future research objective.

Rule induction algorithms typically include a parameter which allows the users to trade generality for accuracy. When such a parameter is set to favor accuracy, more rules may be generated. On the one hand, more rules produce a finer grained ranking, but on the other hand, these rule tend to be over-specific.

4 Empirical Study

We conducted experiments, using six of the data sets of the UCI repository. The data sets are D1: Breast Cancer, D2: Chess (two class scenario), D3: German Credit, D4: Japanese Credit, D5: Magic, and D6: Yeast. The sample sets present a wide range of domains and cover a comprehensive suite of data characteristics. We have utilized Ripper [5] as our rule induction algorithm. For comparison purposes, we have implemented a simple ID3 type of decision tree learning with the Laplace correction (section 3.1.1.) as proposed by Provost and Domingos [11].

The AUC result on each data set was measured via a 10-fold cross validation. Runs with three separate groups of rule sets were carried out on identical data sets. The three groups of rules represented different levels of simplifications for the generated rule sets (i.e., a larger number of more specific and precise rules or a smaller number of more general and overlapping rules) and were obtained by changing the Ripper parameter settings.

Six different scoring methods were used to compute the score of each example. The first three are based on the probability estimation with different methods, *Max*, *Average*, and *Probabilistic Sum*, for combining the scores of an example covered by more than one rule, as describe in Section 3.4. These three methods are denoted as follows.

- *AUC-FR0*: Max
- *AUC-FR1*: Average
- *AUC-FR2: Probabilistic Sum*

The remaining group of three methods represents the use of the Laplace corrections with three score combining methods. They are denoted as:

- *AUC-LC0*: Max
- *AUC-LC1*: Average
- *AUC-LC2: Probabilistic Sum*

Figure 4 depicts the summary of AUCs for the six different methods. AUCs are the average on all six data sets. In the figure, Series1 is the rule set with the more general rules, Series2 is the rule set with less general rules, and Series3 is the rule set with the most specific rules. Except for FR0, rules of Series2 and Series3 outperformed rules of Series1. Namely, specific rules outperformed general rules. This suggests that specific rules are better for ranking than general rules, since specific rules are able to produce a finer grained ranking than the general rules. However, rules of Series2 and Series3 performed about the same. It is also shown in the figure that *Probabilistic Sum* performed better than *Average*, which in turn did better than *Max*. Again, this is because *Probabilistic Sum* and *Average* produce a finer grained ranking than *Max* and assume that examples covered by multiple rules should be ranked higher. Laplace correction achieved about the same performance as the simple probability estimation.

Tables 1 gives the detailed AUC numbers for each of the six data sets. The *R* measurement represents a ratio of the number of perfect rules (rules that cover no negative examples) to the total number of rules in a given rule set. Here the average value of *R* is depicted for each of the three runs of six data sets. The *R* measurement is affected by using Ripper's *S* parameter which simplifies or specializes the generated rule set (i.e., trading off generality for accuracy by the degree of hypothesis simplification). When the parameter is set to favor generation of more specific rules, more accurate results can be achieved.

Tables 2 gives the AUC numbers using the decision tree learning with ranking capability for each of the six data sets.

The following is a summary of the observations drawn from the results:

- The decision tree ranking approach results in a lower AUC performance than the rule-based ranking approaches.
- Specific rules produce higher AUCs than general rules.
- *Probabilistic Sum* generates the best AUC results in comparison with *Average* and *Max*.
- Simple probability estimation method performs about the same as Laplace correction.
- Specific rules produce more scores than general rules.
- AUC correlates and increases with the number of scores.

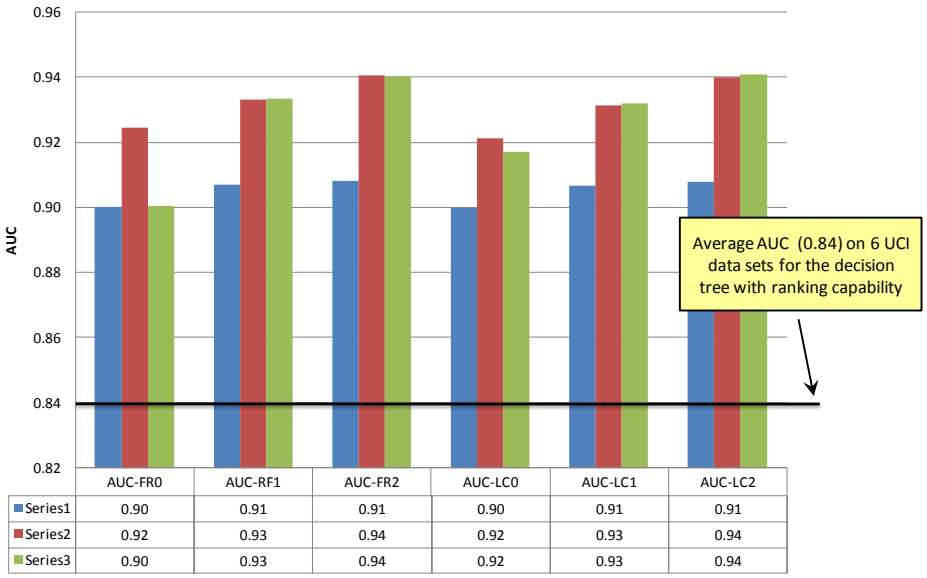

Fig 4. Graphical summary of the AUC experimental results

Table 1. Summary of three series of AUC experimental results.

Dataset	AUC-FR0	AUC-RF1	AUC-FR2	AUC-LC0	AUC-LC1	AUC-LC2
Series 1	**R=0.19**					
D1	0.98	0.98	0.98	0.98	0.98	0.98
D2	0.77	0.79	0.79	0.77	0.78	0.79
D3	0.92	0.92	0.92	0.92	0.92	0.92
D4	0.82	0.84	0.84	0.82	0.84	0.84
D5	1.00	1.00	1.00	1.00	1.00	1.00
D6	0.91	0.91	0.91	0.91	0.91	0.91
Averages	**0.90**	**0.91**	**0.91**	**0.90**	**0.91**	**0.91**
Series 2	**R=0.25**					
D1	0.99	0.99	0.99	0.99	0.99	0.99
D2	0.84	0.85	0.88	0.83	0.85	0.87
D3	0.95	0.95	0.96	0.95	0.95	0.96
D4	0.85	0.88	0.89	0.84	0.87	0.89
D5	1.00	1.00	1.00	1.00	1.00	1.00
D6	0.92	0.93	0.93	0.92	0.93	0.93
Averages	**0.92**	**0.93**	**0.94**	**0.92**	**0.93**	**0.94**
Series 3	**R=0.28**					
D1	0.92	0.99	0.99	0.95	0.99	1.00
D2	0.84	0.85	0.87	0.83	0.85	0.87
D3	0.96	0.96	0.96	0.95	0.95	0.96
D4	0.85	0.88	0.89	0.84	0.87	0.89
D5	0.90	1.00	1.00	1.00	1.00	1.00
D6	0.93	0.93	0.93	0.93	0.93	0.93
Averages	**0.90**	**0.93**	**0.94**	**0.92**	**0.93**	**0.94**

Table 2. Summary of AUC experimental results for the decision tree with ranking capability

D1	0.948736
D2	0.779182
D3	0.891147
D4	0.722007
D5	0.999982
D6	0.705955
Averages	**0.84**

5 Conclusions

In this paper, we have proposed a framework for ranking with rules. The framework introduces three types of rule ranking methods: post analysis of rules, hybrid methods, and multiple rule set analysis. We have also proposed three methods: *Max*, *Average*, and *Probabilistic Sum*, for combining the scores of multiple rules. We have studied the impact of rule learning bias on the ranking performance. An empirical study has been conducted to evaluate the two simplest rule ranking methods, probability estimation with and without Laplace correction, the three rule score combining methods, and the impacts of rule induction bias. The results have been compared with those reported for a decision tree with ranking capability. The comparison shows a lower performance for the decision tree learning approach.

The experimental results clearly indicate that inductive bias has an impact on the performance of ranking. It is also shown that *Probabilistic Sum* and *Average* perform better than *Max*. It seems that a method that produces more scores usually outperforms a method that produces fewer scores. More experiments need to be conducted to verify this claim.

In future research, we will empirically validate other methods of the framework, specifically the ones that use geometric measures. With geometric methods, more scores could be produced. It would be interesting to see whether such geometric measures could indeed improve the AUC or not.

References

1. Alvarez I. & Bernard S. (2005). Ranking Cases with Decision Trees: a Geometric Method that Preserves Intelligibility. IJCAI, 635-640.
2. Barsalou, (1985). Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories. In Journal of Experimental Psychology: Learning, Memory and Cognition, 11:629-654.
3. Bauer, E. & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning, 36: 105-139.
4. Buchanan & Shortliffe (eds.) (1984). Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.

5. Cohen, W. (1995). Text categorization and relational learning. In ICML 1995: 124-132. 1995.
6. Cortes, C. & Mohri, M. (2003). AUC optimization vs. error rate minimization. In Advances in Neural Information Processing Systems (NIPS'03). MIT Press.
7. Ferri C., Flach, P.A., & Hernandez-Orallo, J. (2003). Improving the AUC of probabilistic estimation trees. In Proceedings of the Fourteenth European Conference on Machine Learning. Springer.
8. Fürnkranz, J. (2002). Round Robin Classification. Journal of Machine Learning Research, 2: 721-747.
9. Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.
10. Ling C.X. & Yan, R.J. (2003). Decision tree with better ranking. In Proceedings of the Twentieth International Conference on Machine Learning. Morgan Kaufmann.
11. Provost F. J. & Domingos, P. (2003). Tree induction for probability-based ranking. Machine Learning, 52(30).
12. Waterman A., Donald., (1986). A Guide to Expert Systems. Reading, Mass (USA). Addison-Wesley Publishing Company. pp 49-60.
13. Wang L.X. & Mendel J.M. (1992), Generating Fuzzy Rules by Learning from Examples, IEEE Transactions on Systems, Man and Cybernetics, Volume 22, No 6., pp. 1414-1427.
14. Zhang J & Michalski, R.S. (1995). An Integration of Rule Induction and Exemplar-Based Learning for Graded Concepts. Machine Learning 21(3): 235-267.