Ranking - Methods for Flexible Evaluation and Efficient Comparison of Classification Performance

Saharon Rosset

Department of Statistics, Tel Aviv University
P.O. Box 39040 Tel Aviv, Israel

Abstract

We present the notion of Ranking for evaluation of two-class classifiers. Ranking is based on using the ordering information contained in the output of a scoring model, rather than just setting a classification threshold. Using this ordering information, we can evaluate the model's performance with regard to complex goal functions, such as the correct identification of the k most likely and/or least likely to be responders out of a group of potential customers. Using Ranking we can also obtain increased efficiency in comparing classifiers and selecting the better one even for the standard goal of achieving a minimal misclassification rate. This feature of Ranking is illustrated by simulation results. We also discuss it theoretically, showing the similarity in structure between the reducible (model dependent) parts of the Linear Ranking score and the standard Misclassification Rate score, and characterizing the situations when we expect Linear Ranking to outperform Misclassification Rate as a method for model discrimination.

1. Introduction

A standard classification model is a function c, which gets a data point x as input and returns the expected class for this data point, c(x). In the 2 class case we have $c(x) \in \{0,1\}$.

A simple scheme for building a classifier for a given problem involves 2 main stages: (1)Based on a training set of cases with known outcomes, using one or more modeling techniques (algorithms), create a number of good candidate models, and (2) Choose a good model (preferably the best) by evaluating the candidate models on a test set of "new" cases with known outcomes and comparing their performance.

This paper discusses the second stage in 2-class (0/1) problems.

In standard 0/1 situations the most common evaluation technique is the misclassification (error) rate, which simply counts the number of erroneous decisions made by c(x) on the test set. This technique gives reliable predictions of future performance of the classification model.

A major shortcoming of the misclassification rate is its limited ability to reflect the goal of the classification or decision process. For example, when misclassification costs are highly unbalanced or the distribution of the classes is highly skewed and unstable, the misclassification rate can be inadequate (Provost and Fawcett (1997) describe this problem and suggest an alternative).

In some cases, the goal of the process can be more specific and include more than just 0/1 decisions - like locating the k customers out of n who are most likely to respond in a direct mailing campaign. Such goals extend the definition of "classification model" a little, but are a realistic description of actual needs. Evaluation of classification models with such goals in mind requires an adequate, specialized evaluation method.

This paper describes Ranking, a family of evaluation techniques which are flexible in describing the goal of the process and allow the researcher to adjust the evaluation technique to the required goal. Some examples are given to illustrate this flexibility.

The main result presented here, however, is that even in the simplest case , 0/1 decisions with equal misclassification costs, the Misclassification Rate is not an efficient technique for comparing classifiers and choosing the better one. It is shown that a member of the Ranking family, Linear Ranking, significantly exceeds misclassification rate as a means of discriminating between classifiers and identifying the better one. This is shown in simulation results and discussed theoretically.

2. Ranking

Many of the common algorithms for building 2-class classifiers, like Naive Bayes, K-NN and Logistic Regression, actually estimate $Pr(y=1|x)$, the probability that an unseen case belongs to class "1". The actual classification is then performed by comparing this estimate to a given threshold (usually 0.5 in the equal misclassification cost setting).

Friedman (1997) notes that even for good classifiers these probability approximations may be highly biased and thus ineffective as probability estimates. However, they may still be treated as scores rather than probability approximations, and as such their ordering may contain a lot more information than is available by just thresholding them into dichotomous decisions.

In fact, some other common modeling techniques which do not give probability approximations, such as Linear

Discriminant Analysis, Linear Regression (Hastie, Tibshirani and Buja 1994) and Back-Propagation Networks (Rumelhart, Hinton and Williams 1986) can also be viewed as creating scoring models rather than just binary decision boundaries.

The information contained in the ordering of the scores can be used both to address more complex needs than just 0/1 decisions, and to obtain better evaluations for the standard 0/1 decision task.

From now, and through the next sections, we assume we have a scoring model m(x) and a test set $\{x_i,y_i\}$ for $i\in\{1...n\}$. Let h be the permutation which describes the ordering of the test set observations by their scores (as given by the model): $m(x_{h(1)}) \le m(x_{h(2)}) \le \ ... \ \le m(x_{h(n)})$

A Ranking score for the model on this test set will be defined as:

$$s(m) = \sum_{i=1}^{n} g(i) I\{y_{h(i)} = 1\}$$

where $g:N\rightarrow R$ is the selected monotone non-decreasing Ranking function and I is the standard indicator function.

It is easy to see how Ranking can be formulated for Bootstrap and Leave-Out-Many Cross Validation as well. We limit the discussion in this paper to test-set situations.

3. Examples of Ranking Scores

Following are some specific examples of Ranking scores. These examples are meant to demonstrate the flexibility of the family of Ranking scores and its ability to reflect and represent different research goals. Detailed discussion of the matching of Ranking functions to specific problems is outside the scope of this paper.

1. Misclassification rate as a Ranking score. A lower bound on the number of misclassifications on the test set can be obtained as a ranking score by setting: $g(i) = I\{i > n_0\}$.

Then we get $s(m) = n_1 - e/2$ with e the minimal number of errors which any threshold would achieve for this model. (n_0 and n_1 are, respectively, the numbers of zeros and ones in the test set.)

2. Linear Ranking. Linear Ranking is based on setting $g(i)= i$. We then get:

$$s(m) = \sum_{i=1}^{n} i \cdot I\{y_{h(i)} = 1\}$$

In some cases, Linear Ranking may be a good approximation of the researcher's real goal, like in the case of direct mailing campaigns, where the goal could be to successfully rank the potential customers rather than give them 0/1 results.

In section 5 we show that the Linear Ranking score has a symmetry feature - if we switch the class labels, comparison of two classifiers' scores still gives the same results. This symmetry implies that Linear Ranking offers an evaluation with equal "misclassification costs" for the 2 kinds of errors (although different costs for different "magnitude" errors).

Sections 4-7 discuss the use of Linear Ranking as an evaluation and comparison method for standard classification problems. They illustrate and discuss Linear Ranking's increased effectiveness in identifying the better classification model, compared to the Misclassification Rate, in the case of equal misclassification costs.

3. Quadratic Ranking. Set $g(i) = i^2$. The resulting score gives higher penalties to errors in the higher segments of the sorted test set, and creates a non-balanced score which gives more emphasis to getting the order right for the cases with the higher probability of being "1". This may be relevant when the actual goal of the researcher is to correctly identify the cases (customers) who are most likely to belong to class 1, and correct ordering among the cases less likely to belong to class 1 is of secondary importance. There are, of course, many other Ranking scores which would put the emphasis on ordering the likeliest cases correctly. Choosing the right one could be an important issue, and is very relevant to successful and efficient use of the idea of Ranking for model evaluation in such situations.

4. Simulation Results

In this section we describe simulations experimenting with Linear Ranking on 2 of the simplest and most popular algorithms for creating classification models : Naive Bayes and K-nearest neighbors (K-NN). The goal of these simulations is to show that the Linear Ranking score can actually judge the classification performance better than Misclassification Rate in real life situations and algorithms. We show this by creating multiple pairs of models of the same sort and demonstrating that Linear Ranking consistently identifies the better model with a higher probability across different pairs of models and large numbers of randomly generated test sets. This phenomenon is discussed theoretically in the next sections.

For simplicity all the examples in this section were constructed assuming equal prior probabilities for the 2 classes and equal misclassification costs. As both algorithms are actually probability approximation algorithms the threshold for classification is always 0.5.

4.1 Naive Bayes Classifiers

The Naive Bayes algorithm (see Langley, Iba and Thompson 1992) approximates the probability Pr (y=1) in terms of the marginal "conditional" probabilities of the x-vector's components. For the equal priors case it reduces to:

$$m(x) = \hat{\Pr}(y = 1|x) = \frac{\prod_{i=1}^{d} \#\{x_j = x_j \bigcap y = 1\}}{\prod_{i=1}^{d} \#\{x_j = x_j \bigcap y = 1\} + \prod_{i=1}^{d} \#\{x_j = x_j \bigcap y = 0\}},$$

with $x = \{x_j\}_{j=1}^{d}$ the "new" x-vector. The summations on the right side (the # signs) are over all training set cases.

For the simulations a 10-dimensional input vector was used (d=10). All x-values were randomly and independently drawn in [0,1].

The probability of y to be 1 was set to depend only on the first 2 coordinates:

$$\Pr(y=1|x) = f(x_1,x_2) = \tfrac{1}{2} \cdot (x_1 + x_2)^2 \cdot I\{x_1 + x_2 \le 1\} + [1 - \tfrac{1}{2} \cdot (2 - x_1 - x_2)^2] \cdot I\{x_1 + x_2 > 1\}$$

Twenty training sets of size 1000 each were drawn. For each training set, 100 test sets of size 100 each were drawn. The training set x-values were discretized using a standard χ^2 method.

It is easy to see that the real Bayes ("oracle") classifier (which gives y=1 iff $x_1 + x_2 \le 1$) cannot be expressed as a Naive Bayes classifier. The best Naive Bayes classifier should clearly be the one using x_1 and x_2 for the model. Classifiers with only x_1 or x_2 will generally be "underfitted" (lack vital model flexibility) while classifiers with additional x-vectors will generally be "overfitted" (contain flexibility which results from "noise").

The simulation compared the Naive Bayes classifier using only x_1 to the classifier using both x_1 and x_2. Explicitly, the classifiers are:

$$m_1(x) = \frac{\#\{x_1 = x_1 \cap y = 1\}}{\#\{x_1 = x_1\}}$$

$$m_2(x) = \frac{\prod_{i=1}^{2} \#\{x_i = x_i \cap y = 1\}}{\prod_{i=1}^{2} \#\{x_i = x_i \cap y = 1\} + \prod_{i=1}^{2} \#\{x_i = x_i \cap y = 0\}}$$

Thus, two models were created from each of the twenty training sets. These models were used to score the 100 test sets. Each of the models was evaluated on each of the test sets using both Misclassification Rate and Linear Ranking. The result which is of interest is the percentage of test sets on which the better model (m_2) gives a better overall evaluation score compared to the lesser model (m_1).

It should be noted that if this percentage had turned out to be not significantly more than 50% for any of the 20 pairs of models it would mean that for that training set m_2 is in fact not better than m_1. This was not the case here, however. Each of the twenty different pairs of models indicated more than half the time that m_2 was better than m_1. The results show that m_2 was always much better than m_1 in terms of future classification performance. The overall average misclassification rates of the 2 kinds of models were: m_1 - 33.5% m_2 - 27.5%.

The results are summed up in table 1. The first row shows the average percentage of test sets on which m_2 was better than m_1 (averaged over the twenty different pairs of models, each pair generated using a different training set). The second row shows the minimal number of times out of a 100 that m_2 was better (the minimum is over the 20 pairs of models) and the third row shows the number of times m_2 got a perfect record over m_1 by being better on all 100 test sets.

These results indicate clearly that Linear Ranking succeeded much better than Misclassification Rate in tracking down the better model - both in the "average" sense (98.6% correct vs. 90% correct) and in the "worst case" sense.

So, by using Linear Ranking we significantly improve our chances of identifying the better model successfully, and making fewer classification errors in the future.

4.2 K-NN Classifiers

K-Nearest Neighbors models approximate the probability Pr(y=1|x) of a new case by the average of its K nearest neighbors in the training set:

$$m(x) = \frac{\#\{y = 1 \text{ of K nearest training neighbors}\}}{K}$$

Different models are created by using different values of K.

The setup here was quite similar to the Naive Bayes simulations - all x-vectors are 10-dimensional independently drawn from U[0,1]. We derived twenty training sets with 1000 cases each and for each training set (and resulting pair of models) there were 100 test sets of 100 cases each.

Here the probability for y to be 1 was simply set as the value of x_1: $\Pr(y=1|x) = x_1$

Friedman (1997) has shown that the K (number of neighbors), which gives the best classification performance (smallest misclassification rate) tends to be large, much larger than the optimum for probability estimation. Simulations with Linear Ranking showed clearly that its behavior is similar to that of classification.

Here, m_1 was chosen with K=10 and m_2 with K=50. Given Friedman's results, we expected m_2 to be better and the results confirmed this conjecture. The overall average misclassification rate of the models : m_1 - 28.5% m_2 - 26.5%

The results can be seen in table 2. The first two rows in the table have similar comparisons to the ones in table 1, while the third row contains a comparison of the number of times (out of the 20) m_2 was better than m_1 on at least 80 of the 100 test sets.

The results here are even more conclusive than for Naive Bayes classifiers - we even have a "threshold" of 80% correct discrimination between the models which

	Misclassification Rate	Linear Ranking
Ave. Pct. of m_2 better than m_1	90.05	98.60
Min. Pct. of m_2 better than m_1	69	95
No. of times m_2 better than m_1 on all 100 test sets	1	8

Table 1: *Results of comparing Naive Bayes classifiers with 1 and 2 predictors by using the Misclassification Rate and Linear Ranking Score criteria.*

	Misclassification Rate	Linear Ranking
Ave. Pct. of m_2 better than m_1	69.72	92.6
Min. Pct. of m_2 better than m_1	61	86
No. of times m_2 better than m_1 on at least 80 test sets	0	20

Table 2: *Results of comparing K-NN classifiers with K=10 and K=50 by using the Misclassification Rate and Linear Ranking Score criteria.*

Linear Ranking achieved on all 20 pairs of models while Misclassification Rate achieved on none.

5. Probabilistic Analysis of Linear Ranking

As before, we assume we have a scoring model, and a test set ordered according to the model's scores. We also assume a set of "true" probabilities p, with $Pr(y_i=1|x_i) = p_i$ for this test set. We further assume now, without loss of generality, that the test set is ordered according to these "true" probabilities: $p_1 \le p_2 \le \ldots \le p_n$

Reducible-Irreducible Decomposition

The first step in understanding how a choice of classifier is related to the evaluation score to be examined is to decompose the score into the part that depends only on the problem (the "irreducible" or more appropriately here "predetermined" part) and the part that is determined by the chosen model (the "reducible" or "model dependent" part). This type of decomposition for estimation and misclassification rate can be found in Friedman (1997).

Following the ideas used by Friedman we define the predetermined part as the score of an "oracle" model with complete knowledge of the real p_i's. The oracle would correctly order the items and attain the score:

$$\sum_{i=1}^{n} i \cdot I\{y_i = 1\} = \sum_{i=1}^{n} i \cdot y_i$$

The decomposition then becomes:

$$\sum_{i=1}^{n} i \cdot y_{h(i)} = \sum_{i=1}^{n} i \cdot y_i - \sum_{i=1}^{n} i \cdot [y_i - y_{h(i)}] \qquad (5.1),$$

with the last term describing the reducible (model dependent) part of the ranking score.

A smaller reducible part means a bigger, hence better, score. The mean of the reducible part is always non-negative (because no classifier can attain a better average result that the "oracle").

Symmetry Feature of the Reducible Part. The reducible part in (5.1) is symmetric, in the sense that if the class labels are switched together with the order of the scores and Linear Ranking is performed again, the reducible part will be the same. In terms of the current notations this reducible part (with switched class labels) will be:

$$\sum_{i=1}^{n} i \cdot [I\{y_{n+1-i} = 0\} - I\{y_{h(n+1-i)} = 0\}] \qquad (5.2)$$

Because of the dichotomous nature of the identity function, (5.2) is equal to:

$$\sum_{i=1}^{n} -i \cdot [I\{y_{n+1-i} = 1\} - I\{y_{h(n+1-i)} = 1\}] = \sum_{i=1}^{n} -i \cdot [y_{n+1-i} - y_{h(n+1-i)}]$$

which can be rearranged as:

$$\sum_{i=1}^{n} (n+1-i) \cdot [y_{n+1-i} - y_{h(n+1-i)}] - (n+1) \cdot \sum_{i=1}^{n} [y_{n+1-i} - y_{h(n+1-i)}]$$

The sum in the second term is zero, and the first term is simply the reducible part in (5.1).

Re-formulation of the Reducible Part. The reducible part of the Linear Ranking score in (5.1) depends on the permutation h(i) relative to the identity permutation. A different representation of the reducible part can be achieved as a function of only the pairwise switches of order between the 2 permutations. This representation will prove very useful in analyzing the Linear Ranking score. The following equivalence is the basis for this representation:

$$\sum_{i=1}^{n} i \cdot (y_i - y_{h(i)}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (y_j - y_i) \cdot I\{m(x_i) > m(x_j)\} \quad (5.3)$$

So the reducible part is the sum of the class differences over all the pairs that switched places between their "true" probability ordering and the ordering determined by the model scores (as described by the permutation h).

The proof of this equivalence is quite simple, based on the representation of the permutation h as a collection of pairwise "switches", but will not be presented here.

6. The Two Model Comparison Procedures

We now want to compare the Misclassification Rate score (MC) and Linear Ranking score (LR) as methods of model discrimination and to gain some insight into why LR might perform better. The decomposition of the scores into reducible and irreducible parts is a key tool.

For MC, (Friedman 1997) formulated the reducible part:

$$\sum_{i=1}^{n} [I\{c(x_i) \neq y_{B,i}\}] \cdot [I\{y_i = y_{B,i}\} - I\{y_i \neq y_{B,i}\}]$$

Or, equivalently: $\sum_{i=1}^{n} I\{m(x_i) \le t\} \cdot (1 - 2y_i) \qquad (6.1),$

with y_B representing the Bayesian ("oracle") decision. In this formulation, as with LR, a smaller reducible part means a better result (as we presented MC here it actually counts the number of correct classification decisions).

Now we can formulate explicitly the expressions for a comparison of 2 classifiers, m_1 and m_2, using MC and LR. Because the irreducible parts are equal for both models, the difference between the models' scores is the difference

between the reducible parts. Using (6.1) and (5.3) we get:

$$MC(m_2) - MC(m_1) = \sum_{i=1}^{n} [I\{m_1(x_i) \le t\} - I\{m_2(x_i) \le t\}] \cdot (1 - 2y_i) \quad (6.2)$$

$$LR(m_2) - LR(m_1) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} [I\{m_1(x_i) > m_1(x_j)\} - I\{m_2(x_i) > m_2(x_j)\}] \cdot [y_j - y_i] \quad (6.3)$$

For either MC or LR, m_2 will be preferred if this difference is positive.

7. Discussion

In comparing LR and MC, equations (6.3), (6.2) show the similarity in their structure.

$I\{m_1(x_i) \le t\} - I\{m_2(x_i) \le t\}$ in (6.2) compares a thresholding decision between the two models, while $I\{m_1(x_i) > m_1(x_j)\} - I\{m_2(x_i) > m_2(x_j)\}$ in (6.3) compares an ordering decision. If these decisions differ, they are confronted with the evidence from the test set, and 1 is added to or subtracted from the sum. For example: for MC, if $m_1(x_i) \le t$ and $m_2(x_i) > t$, the sum will increase by 1 if $y_i=1$ (m_2 was "right") and will decrease by 1 if $y_i=0$ (m_1 was "right").

The MC score (6.2) compares these decisions on the n test-set cases and checks them against the test-set y values when the 2 models disagree. The LR score (6.3) compares decisions on the n(n-1)/2 pairs of cases and checks the y-values when the decisions regarding the ordering of these cases by their scores disagree.

So in general we can say that MC sums over the results of O(n) comparisons and LR sums over the results of $O(n^2)$ comparisons. This gives us a strong sense of the advantage of LR over the MC - it is in fact the reduced variance of the resulting decision, obtained by using more information to make it. On the other hand, the fewer comparisons we make in (6.2) are the ones which actually determine the classification performance of the models, as $MC(m_2)-MC(m_1)$ is actually an unbiased estimator of the average difference in misclassification rates, $E[MC(m_2)- MC(m_1)]$. Thus, the "average" result of (6.2) is guaranteed to be the right one. The comparisons in (6.3) check slightly different attributes of the scoring model. A scoring model may have one behavior with regard to the "threshold crossings" tested in (6.2) and a different one with regard to the "binary switches" tested in (6.3). So, LR may be "biased" in that its average result may not reflect the true difference in classification accuracy. Consequently, models for which LR should be effective as a comparison method would be ones where the scoring behavior is "consistent" in the sense that threshold crossings are as common as can be expected given the number and magnitude of binary order switches and vice versa. When this consistency does not exist we can expect that LR will not always outperform MC in selecting the better classification model.

It is of course easy to build artificial examples where there would be no such consistency and LR's indication as to the better model for classification would be erroneous.

We can also describe families of models for which we can be sure that LR will be a better discrimination method than MC. We are currently working towards formulating generalizations which would set a more formal basis for the use of LR (and hopefully Ranking in general).

Intuitively it seems reasonable that most of the algorithms for creating scoring models, which fit the data to some non-trivial structures, should not display inconsistent behavior between threshold crossings and binary order switches. This indicates that LR should indeed be an efficient discrimination method in most practical situations.

8. Conclusions

We have introduced Ranking as a family of evaluation techniques for scoring models. Ranking methods have two main interesting features:

1. When the goal of the researcher is not strictly minimum error rate classification, it gives a tool for defining a variety of target functions and testing the suggested models against them.
2. For the standard minimum error rate classification task, Ranking can provide a more efficient tool for comparison of suggested models and selection of the best one than the standard evaluation methods (like Misclassification Rate). This was illustrated through the improved efficiency of Linear Ranking in selecting the better classification model compared to Misclassification Rate.

Acknowledgments

I would like to thank David Steinberg of Tel-Aviv University for his counseling and Gadi Pinkas of Amdocs Inc. for introducing me to Ranking.

References

Friedman, J. H. 1997. On Bias, Variance, 0/1 Loss, and the Curse of Dimensionality. *Data Mining and Knowledge Discovery* 1:55-77.

Hastie, T.; Tibshirani, R.; Buja A. 1994. Flexible Discriminant Analysis by Optimal Scoring. *J. of the American Statistical Society*. 428:1255-1270.

Langley P.; Iba, W., and Thompson, K. 1992. An Analysis of Bayesian Classifiers. In *Proceedings of the 10th NCAI*, pp. 223-228. AAAI Press & M.I.T. Press

Provost, F.; Fawcett, T. 1997. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distribution. In *Proceedings of KDD-97*, pp. 43-48. Menlo Park, CA: AAAI Press.

Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press.

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. 1986. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*. Vol. 1: 318-362. M.I.T