Using an Ensemble Classifier for Machine Learning Applications

COSTAS TSATSOULIS AND DANICO LEE
Department of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center (ITTC)
The University of Kansas
2335 Irving Hill Rd., Lawrence, KS 66045
USA
http://www.ittc.ku.edu/~tsatsoul

Abstract: - We describe an architecture for integrating classification algorithms that have been created by a variety of machine learning methods, trained on the same data set. The ensemble classifier unifies all these classifiers into a single module and uses voting and a reward/punishment system to select the best classifier for a specific data set. In this paper we discuss the theory behind the ensemble architecture, and present its implementation and a set of experiments using a variety of data sets. Our work shows how the ensemble performs as well or better than the best classifier for a specific data set on most occasions.

Key-Words: - machine learning, classification algorithms, ensemble classifier

1 Introduction

Individual machine learning algorithms (such as k-nearest neighbor or inductive learning techniques) perform with varying accuracy on different data sets. It is impossible to determine *a priori* which algorithm will perform better on what data set, based on the data set's characteristics (such as missing values, noise, number of attributes, type of attributes, etc.) Some work in machine learning has started identifying types of classification algorithms that may work better under different circumstances, but this work is still at early stages [1].

Another possibility to dealing with the uncertainty of which algorithm to select for a specific data set, is not to make this decision. We propose to incorporate many classifiers that have been trained on the same data set by a variety of machine learning algorithms into a single classifier, an *ensemble classifier*.

Ensemble learning is a machine learning technique that selects a collection, or *ensemble*, of hypotheses from the hypothesis space and combines their predictions. An ensemble consists of a set of independently trained classifiers whose predictions are combined when classifying new instances.

In our ensemble classifier we integrated five classifiers and used a voting mechanism to determine which of the classifiers to believe. The ensemble rewards classifiers that make the correct prediction and punishes those that are wrong. After a training period the ensemble learns to prefer the best classifier for a specific data set.

We performed extensive experiments on a variety of data sets and showed that in most cases the ensemble classifier performed as well as or better than the best classifier for a specific data set. We also showed that the ensemble tended to prefer different classifiers for different data sets. The result is an architecture that successfully integrates various classifiers that have been trained by different machine learning algorithms, and which efficiently selects the best classifier for a specific data set.

2 Ensemble Classifiers

Experiments conducted with a variety of classification algorithms generated by machine learning techniques have shown that the accuracy of a single classification algorithm differs greatly across data sets due to the structural variability of the data and the specific nature of values. It is impossible to predict which classification algorithm will work best for what type of data and domain. To solve this problem, we developed an ensemble classifier. An *ensemble classifier* is a collection of a number of classification algorithms where each of them provides classification predictions. The ensemble learns non-linearly which individual algorithms provide better predictive accuracy for different data domains resulting in a classifier that adapts itself to the specific data set, and performs better than any individual predictive algorithm.

Ensemble learning is a non-linear machine learning technique that selects a collection, or *ensemble*, of hypotheses from the hypothesis space and combines their predictions [1]. An ensemble consists of a set of independently trained classifiers whose predictions are combined when classifying new instances. The one of the most widely used ensemble methods is called *boosting*. Boosting works with a weighted training set. Each example in the training set has an associated weight $w_j >= 0$. The higher the weight of an example, the higher is the importance attached to it during training. Boosting, produces a series of classifiers, where the training set used for each member of the series is chosen based on the performance of the earlier classifier(s) in the series. All examples start with $w_j = 1$ and boosting increases the weights (importance) of misclassified examples and decreases the weights of the classified examples. Therefore, examples that are incorrectly predicted by previous classifiers in the series are chosen more often than examples that were correctly predicted. As a result, boosting produces new classifiers for its ensemble that are better able to correctly predict examples for which the current ensemble performance is poor [2].

2.1 Suggestion Aggregator - Voting Among Internal Classifiers

Since the ensemble receives predictions from many internal classifiers, voting is used for aggregating these inputs into a single suggestion [3]. The voting system forms a consensus decision on which value is suggested by most of the classifiers. All classifiers return the same number of maximum classifications. Each classifier's suggestions are ranked; if a classifier returns N suggestions the top one receives a value of N, the second one N-1, and so on, with the last suggestion receiving a 1. In addition to their rank, suggestions are modified by the weight of their classifier. Initially all classifiers have the same weight, but this is modified based on which classifier works better for the specific data set domain (more on this later). An example of the voting mechanism is shown on Figure 1. In this example, three classifiers provide a maximum of three suggestions each. Classifier A makes three suggestions; the top one receives a rank value of 3, the second one of 2, and the third one of 1; the rank values are multiplied by the weight of the classifier (0.67) and then normalized by the sum of the weights of all the classifiers. The same occurs for the suggestions by the other classifiers. The suggestion with the highest support is the one selected by the ensemble and presented to the user.

2.2 Ensemble Learning – Weighting Each Classifier by Past Performance

The weighting system in our ensemble learns from past performance of the internal classifiers. As classification algorithms exhibit different predictive accuracy for a specific data set, the overall accuracy of the suggestion system is improved by learning how each internal classifier performs on each data set and weighting their votes accordingly.

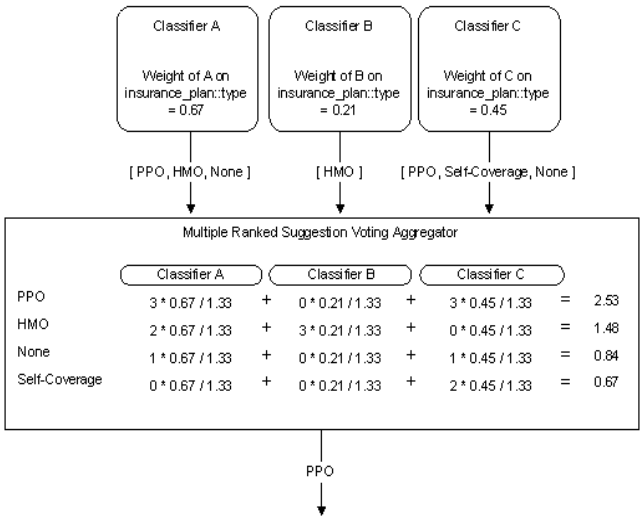

Figure 1: Ensemble classifier: individual, internal classifiers make predictions which are ranked, with the rank multiplied by the classifier weight (confidence). The suggestion with the highest total support is presented to the user.

Our measure of predictive performance of a classifier is the *weighted accuracy rate* of its suggestions for that data set. The weighted accuracy rate is defined as follows: since each classifier suggest N possible classes with a specific rank (where the higher rank indicates that the classifier has a stronger certainty of the accuracy of the classification), the weighted accuracy rate incorporates this rank into its computation of the overall accuracy rate of a classifier. For example, if a classifier makes three suggestions for a class, C1, C2, C3, so ranked, and the correct class is C2, then the classifier is awarded a 0.66 weighted accuracy. If the correct class were C1 the classifier would have been awarded a 1.0; if the correct class were C3 it would receive 0.33; otherwise it would receive a zero for accuracy. Mathematically:

$$wa_i = \begin{cases} r_i \times \frac{1}{N} \text{ ; if } c \in C_i \\ 0 \text{ ; otherwise} \end{cases}$$

Where wa_i is the weighted accuracy for classifier i, r_i is the rank of the correct solution for that classifier, N is the maximum number of classes returned by the internal classifiers, c is the correct solution, and C_i is the solution set returned by classifier i. (Note that the top class returned is of rank N, the second one rank N-1, etc.)

Since classifiers that make fewer suggestions may be favored (fewer suggestions imply fewer errors), the weight of each predictor is normalized by dividing it by the sum of all weights for all internal classifiers.

$$\text{Normalized} = wt_i = \frac{wt_i}{\sum_{i=1}^{c} wt_i}$$

where wt_i is the weight assigned to the i^{th} classifier and c is the total number of internal classifiers.

Initially the classes suggested by each internal classifier are weighted equally. The internal classifiers that suggested the correct class get rewarded by using the weighted accuracy metric. The internal classifiers get punished if they suggest the wrong class. As a result, our system adapts to the specific domain of the data set and increases its prediction performance.

3 Experiments

In our experiment we used five simple classification algorithms inside an ensemble which learned predictor weights for each data set domain as described above. Each classifier used a different machine learning technique, and four of them were capable of returning multiple possible classes with a ranking.

The first classifier we used was based on Naïve Bayes theory. Naïve Bayes analyzes the relationship between each independent and dependent variable to derive a conditional probability for each relationship. When a new case is analyzed, a prediction is made by combining the effects of the independent variables on the dependent variable (the outcome that is predicted). The Naïve Bayesian Classifier computes the probability of a value for an empty node, based on the conditional probabilities of the predicted value given the actual values of the nodes that have been filled in. In other words, given the values of *i-1* fields: v_1, v_2, …, v_{i-1}, and given a possible v_i for the value of the empty node n_i, the Naïve Bayesian Classifier computes the probability that v_i is predictably correct as follows:

$$P(n_i = v_i) = \frac{\prod_{k=1}^{i-1} P(n_i = v_i \mid n_k = v_k)}{\prod_{k=1}^{i-1} P(n_i = v_i \mid n_k = v_k) + \prod_{k=1}^{i-1} \left(1 - P(n_i = v_i \mid n_k = v_k)\right)}$$

The second and third classifiers were K-Nearest-Neighbor, where K=1 and K=3. The fourth classifier was based on frequency and suggested the most frequently used class. The final predictor simply suggested the most recently found class.

We selected ten data sets from the University of California at Irvine's Machine Learning Repository [4]. The data sets were selected so as to span the size and complexity dimensions. Their size ranged from 24 to 20,000 samples, and they had between 4 and 64 attributes per sample.

We also varied the number of classes returned by the classifiers to evaluate how increasing or decreasing the size of the set of potential classifications affects the weighted accuracy function, and thus affects the weight of the individual internal classifiers.

For each data set, the ensemble was trained on a randomly selected subset of the data, and then tested on the rest of the set.

Finally, we collected the accuracy of the ensemble during testing, as well as the accuracy of each of the internal classifiers for comparison.

4 Results

Figures 2-5 show the classification performance of the ensemble versus the individual classifiers for four data sets. The y-axis is the classification accuracy during testing, while the x-axis is the number of classification suggestions returned by each internal classifier.

Figure 2 shows the results with a very large data set (over 5,600 samples) that had many classes. The ensemble performs as well as or better than the best individual classifier. Note that there seems to be

little difference in performance based on the number of total suggestions returned by the classifiers.

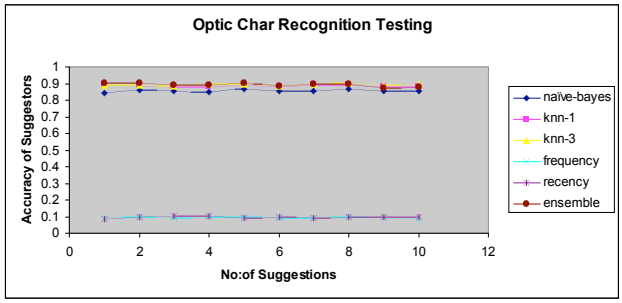

Figure 2: Comparison of ensemble classifier with individual ones for the Optical Character data set from UC, Irvine's Machine Learning Repository.

Figure 3 shows the performance of the system on a smaller data set (90 samples). The ensemble again performs better than the best individual classifier, and this time increasing the number of classes returned improves overall classification accuracy for the ensemble.

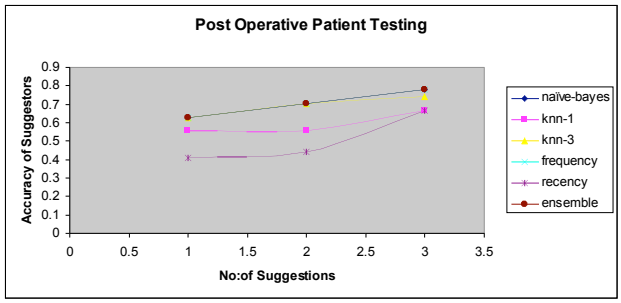

Figure 3: Comparison of ensemble classifier with individual ones for the Post Operative Patient data set from UC, Irvine's Machine Learning Repository.

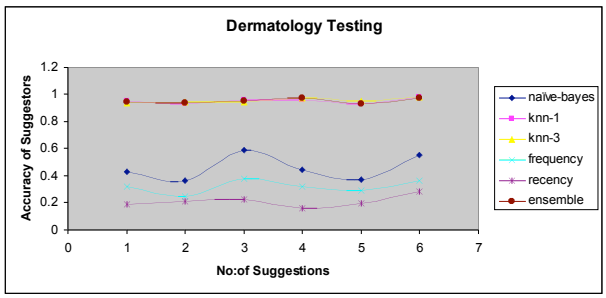

Figure 4: Comparison of ensemble classifier with individual ones for the Dermatology data set from UC, Irvine's Machine Learning Repository.

Figure 4 shows the performance of the system on a medium-sized data set (366 samples). In this example the ensemble selects the best classifier (K-NN with K=3) and almost perfectly follows its performance. This is a dramatic example of how the ensemble can determine the best classifier for a specific data set.

Figure 5 shows the performance of the system on a medium-sized data set (625 samples). In this example the ensemble performs worse than the best individual classifier (Naïve Bayes), but its performance is comparable to the best classifier. Again increasing the number of classes returned improves overall classification accuracy for the ensemble, but not linearly.

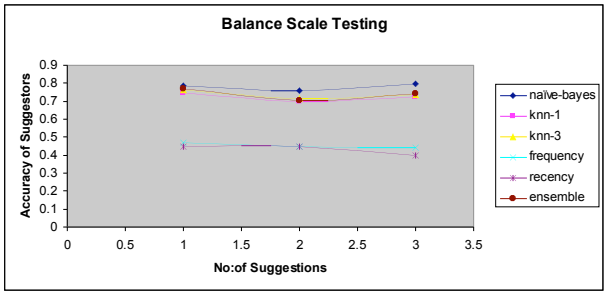

Figure 5: Comparison of ensemble classifier with individual ones for the Balance Scale data set from UC, Irvine's Machine Learning Repository.

Overall, the ensemble classifier performed better or as well as the best internal classifier for seven of the ten data sets we used in our experiments. In two experiments its performance was worse, and in one its performance was very close to that of the best algorithm.

In general, increasing the number of classes returned by each classifier improved the overall classification accuracy.

5 Conclusions

We described the ensemble classifier, an architecture that successfully integrates various classifiers that have been trained by different machine learning algorithms. This ensemble uses voting to select between the classifications suggested by each classifier. It uses a weighted accuracy metric to assign a weight to each classifier, based on how often it makes correct or incorrect predictions as to the class of an unknown sample. The ensemble was tested on ten standard machine learning data sets, and showed to perform better or equally well as the best individual classifier 7 times, and only twice was its performance significantly worse.

The ensemble classifier allows users not to have to worry about which classifier to use for a specific data set, something that can only be established by extensive testing. Instead, a user can integrate any number of classifiers into the ensemble, each classifier generated by a different machine learning algorithm. Then the ensemble will automatically select which algorithm is best, and enable good classification results over any data set.

Acknowledgements:
This work was supported in part by the Kansas Technology Enterprise Corporation.

References:

[1] Russell, S. and Norvig, P. *Artificial Intelligence A Modern Approach*, Second Edition. Prentice Hall, 664-668.

[2] Maclin, R. and Opitz, D. An Empirical Evaluation of Bagging and Boosting, *Fourteenth National Conference on Artificial Intelligence (AAAI-97),* AAAI Press, 1997, 546-551.

[3] G. Weiss, (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

[4] http://www.ics.uci.edu/~mlearn/MLRepository.html (last accessed April 15, 2005).