IOSR Journal of Computer Engineering (IOSRJCE)
ISSN : 2278-0661 Volume 1, Issue 4 (May-June 2012), PP 27-32
www.iosrjournals.org

A Review on Diverse Ensemble Methods for Classification

Prachi S. Adhvaryu[1], Prof. Mahesh Panchal[2]
[1]PG Student, [2](Head of the Computer engineering)
Kalol Institute of Technology

ABSTRACT: *Ensemble methods for different classifiers like Bagging and Boosting which combine the decisions of multiple hypotheses are some of the strongest existing machine learning methods. The diversity of the members of an ensemble is known to be an important factor in determining its generalization error. DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples), that directly constructs diverse hypotheses using additional artificially-constructed training examples. The technique is a simple, general meta-learner that can use any strong learner as a base classifier to build diverse committees. The diverse ensembles produced by DECORATE are very effective for reducing the amount of supervision required for building accurate models. DECORATE ensembles can also be used to reduce supervision through active learning, in which the learner selects the most informative examples from a pool of unlabeled examples, such that acquiring their labels will increase the accuracy of the classifier.*

KEYWORDS: *Data Classification, Ensemble of classifiers Accuracy of classifier, Diversity*

I. INTRODUCTION

One of the major advances in inductive learning in the past decade was the development of ensemble or committee approaches that learn and retain multiple hypotheses and combine their decisions during classification [4].For example, Boosting is an ensemble method that learns a series of "weak" classifiers each one focusing on correcting the errors made by the previous one; and it is currently one of the best generic inductive classification methods.

In this, the new method for generating ensembles, DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples), is presented.
Here, the main concept of DECORATE algorithm is used to predict numeric attribute, as it is only invention for the nominal-categorical attribute. In the weka, it is shown that the DECORATE algorithm is good for the nominal attribute, as compared to bagging and boosting algorithm for the same data.

II. Basic concept of Data Mining

Data mining process (the analysis step of the knowledge discovery in databases process, or KDD), a field of computer science is the process of discovering new patterns from large data sets involving methods such as artificial intelligence, machine learning, statistics and database systems. The goal of data mining is to extract knowledge from a data set in a human-understandable structure and involves database and data management, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of found structure, visualization and online updating.

The following steps are used to preprocess the large dataset.

- Selection: Obtain data from various sources.
- Preprocessing: Cleanse data.
- Transformation: Convert to common format. Transform to new format.
- Data Mining: Obtain desired results.
- Interpretation/Evaluation: Present results to user in meaningful manner.
- Knowledge presentation: where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

Databases are rich with hidden information that can be used for making intelligent business decisions. Classification and prediction are two forms of data analysis which can be used to extract models describing important data classes or to predict future data trends. Whereas classification predicts categorical labels (or discrete values), prediction models continuous-valued functions.

Data classification is a two step process In the first step, a model is built describing a predetermined set of data classes or concepts. The model is constructed by analyzing database tuples described by attributes. Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute. In the context of classification, data tuples are also referred to as samples, examples, or objects. The

data tuples analyzed to build the model collectively form the training data set. The individual tuples making up the training set are referred to as training samples and are randomly selected from the sample population.
Since the class label of each training sample is provided, this step is also known as supervised learning (i.e., the learning of the model is 'supervised' in that it is told to which class each training sample belongs). It contrasts with unsupervised learning (or clustering), in which the class labels of the training samples are not known, and the number or set of classes to be learned may not be known in advance.

2. 1Classifier and its accuracy

Estimating classifier accuracy is important in that it allows one to evaluate how accurately a given classifier will correctly label future data, i.e., data on which the classifier has not been trained. For example, if data from previous sales are used to train a classifier to predict customer purchasing behavior, we would like some estimate of how accurately the classifier can predict the purchasing behavior of future customers. Accuracy estimates also help in the comparison of different classifiers.
Using training data to derive a classifier and then to estimate the accuracy of the classifier can result in misleading over-optimistic estimates due to overspecialization of the learning algorithm (or model) to the data. Holdout and cross-validation are two common techniques for assessing classifier accuracy, based on randomly-sampled partitions of the given data.

2.1.1 Holdout Method

In the holdout method, the given data are randomly partitioned into two independent sets, a training set and a test set. Typically, two thirds of the data are allocated to the training set, and the remaining one third is allocated to the test set. The training set is used to derive the classifier, whose accuracy is estimated with the test set .The estimate is pessimistic since only a portion of the initial data is used to derive the classifier. Random subsampling is a variation of the holdout method in which the holdout method is repeated k times. The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.

2.1.2 Cross-validation

In k-fold cross validation, the initial data are randomly partitioned into k mutually exclusive subsets or folds", s1, s2,…sk, each of approximately equal size. Training and testing is performed k times. In iteration i, the subset Si is reserved as the test set, and the remaining subsets are collectively used to train the classifier. That is, the classifier of the first iteration is trained on subsets S2,…, Sk, and tested on S1; the classifier of the section iteration is trained on subsets S1; S3; ::; Sk, and tested on S2; and so on. The accuracy estimate is the overall number of correct classifications from the k iterations, divided by the total number of samples in the initial data. In stratified cross-validation, the folds are stratified so that the class distribution of the samples in each fold is approximately the same as that in the initial data.

2.1.3 Bootstrap

Other methods of estimating classifier accuracy include bootstrapping, which samples the given training instances uniformly with replacement, and leave-one-out, which is k-fold cross validation with k set to s, the number of initial samples. In general, stratified 10-fold cross-validation is recommended for estimating classifier accuracy (even if computation power allows using more folds) due to its relatively low bias and variance.

III. Increasing classifier accuracy

Bagging (or boostrap aggregation) and boosting are two such techniques. Each combines a series of T learned classifiers, C1; C2;…..CT, with the aim of creating an improved composite classifier,C*.

3.1 Bagging

Given a set S of s samples, bagging works as follows. For iteration t (t = 1,2,…T), a training set St is sampled with replacement from the original set of samples, S. Since sampling with replacement is used, some of the original samples of S may not be included in St, while others may occur more than once. Each bootstrap sample Si contains approx. 63.2% of the original training data. Remaining (36.8%) are used as test set. A classifier Ct is learned for each training set, St. To classify an unknown sample, X, each classifier Ct returns its class prediction, which counts as one vote. The bagged classifier, C*, counts the votes and assigns the class with the most votes to X. Bagging can be applied to the prediction of continuous values by taking the average value of each vote, rather than the majority.

advantages

Bagging works well if the base classifiers are unstable.
It Increased accuracy because it reduces the variance of the individual classifier.

Bagging seeks to reduce the error due to variance of the base classifier.
Noise-tolerant, but no t so accurate

3.2 Boosting

In boosting, weights are assigned to each training sample. A series of classifiers is learned. After a classifier Ct is learned, the weights are updated to allow the subsequent classifier, Ct+1, to "pay more attention" to the misclassification errors made by Ct. The final boosted classifier, C_, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy. The boosting algorithm can be extended for the prediction of continuous values.

advantage

Boosting tends to achieve more accuracy than bagging
Boosting focuses on misclassified tuples so it risks overfitting

limitation

Boosting can fail to perform well given insufficient data. This observation is consistent with the Boosting theory. Boosting also does not perform well
when there is a large amount of classification noise (i.e. training and test examples with incorrect class labels).Boosting is also very susceptible to noise in the data.

Comparison between Bagging and Boosting

Bagging is noise-tolerant, produce better class probability estimates. It is not so accurate. It is related to random subsampling.
While Boosting is very susceptible to noisy data, produces rather bad class probability estimates. It is related to windowing.

IV. Creating Diverse Ensemble Classifier- Diversity

Constructing a diverse committee in which each hypothesis is as different as possible, while still maintaining consistency with the training data, is known to be a theoretically important property of a good ensemble method. [1]

The DECORATE algorithm is used for diverse ensemble method. The DECORATE can also be effectively used for the following:[2]

• Active learning, to reduce the number of training examples required to learn an accurate model;
• Exploiting unlabeled data to improve accuracy in a semi-supervised learning setting;
• Combining both active and semi-supervised learning for improved results;
• Obtaining improved class membership probability estimates, to assist in cost-sensitive decision making;
• Reducing the error of regression methods; and
• Improving the accuracy of relational learners.

Here is some notation and definitation for the supervised learning task.
C is a classifier, a function from objects to classes.
C* is an ensemble of classifiers.
Ci is the ith classifier in ensemble C*.
n is the number of classifiers in ensemble C*.
xi is the description of the ith example/instance.

In supervised learning, a learning algorithm is given a set of training examples of the form {(x1, y1),...., (xm, ym)} for some unknown function y = f(x). The description xi is usually a vector of the form < xi,1, xi,2, ..., xi,k > whose components are real or discrete (nominal) values, such as height, weight, age, eye-color, and so on. The classifier is a hypothesis about the true (target) function f. Given a new example x, the classifier predicts the corresponding y value. The aim of the classification task is to learn a classifier that minimizes the error in predictions on an independent test set
of examples (generalization error). For classification, the most common measure for error is the 0/1 loss function, given by:

$$error_{C,f}(x) = \begin{cases} 0: \text{ if } C(x) = f(x) \\ 1: \text{otherwise} \end{cases}$$

An ensemble (committee) of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. The committee is also reffered by different names in literature-mixtures of experts, classifier ensemble etc.[5]

4.1 Diversity and Error reduction

In an ensemble, the combination of the output of several classifiers is only useful if they disagree on some inputs. We refer to the measure of disagreement as the diversity/ambiguity of the ensemble. For regression problems, mean squared error is generally used to measure accuracy, and variance is used to measure diversity. The generalization error, E, of the ensemble can be expressed as E = □E − □D; where □E and □D are the mean error and diversity of the ensemble respectively. This result implies that increasing ensemble diversity while maintaining the average error of ensemble members, should lead to a decrease in ensemble error. Unlike regression, for the classification task the above simple linear relationship does not hold between E, □E and □D . But there is still strong reason to believe that increasing diversity should decrease ensemble error.[4]

The ensemble diversity is critical to error reduction. The ensemble diversity is compared with the ensemble error reduction, i.e. the difference between the average error of the ensemble members and the error of the entire ensemble. We found that the correlation coefficient between diversity and ensemble error reduction is 0.6602 which is fairly strong. Furthermore, we compared diversity with the base error reduction, i.e. the difference between the error of the base classifier and the ensemble error. The base error reduction gives a better indication of the improvement in performance of an ensemble over the base classifier. The correlation of diversity versus the base error reduction is 0.1607.

We use the disagreement of an ensemble member with the ensemble's prediction as a measure of diversity. More precisely, if Ci(x) is the prediction of the i-th classifier for the label of x; C*(x) is the prediction of the entire ensemble, then the diversity of the i-th classifier on example x is given by

$$d_i(x) = \begin{cases} 0: \text{ if } C_i(x) = C^*(x) \\ 1: \text{otherwise} \end{cases}$$

To compute the diversity of an ensemble of size n, on a training set of size m, we average the above term[2]:

$$\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} d_i(x_j)$$

This measure estimates the probability that a classifier in an ensemble will disagree with the prediction of the ensemble as a whole.

DECORATE differs from ensemble methods, such as Bagging, in that it explicitly tries to foster ensemble diversity. There have been some other attempts at building ensembles that focus on the issue of diversity. To the best of our knowledge, DECORATE is the only method that uses artificially constructed examples to improve generalization accuracy.

To classify an unlabeled example, x, we employ the following method. Each base classifier, Ci, in the ensemble C* provides probabilities for the class membership of x. If $P_{*i},y(x)$ is the probability of example x belonging to class y according to the classifier Ci, then we compute the class membership probabilities for the entire ensemble as:

$$P_y(x) = \frac{\sum_{C_i \in C^*} P_{C_i,y}(x)}{|C^*|}$$

where Py(x) is the probability of x belonging to class y.

4.2 Construction of Artificial Data

For a numeric attribute, we compute the mean and standard deviation from the training set and generate values from the Gaussian distribution defined by these. For a nominal attribute, we compute the probability of occurrence of each distinct value in its domain and generate values based on this distribution. We use Laplace smoothing so that nominal attribute values not represented in the training set still have a non-zero probability of occurrence. In constructing artificial data points, we make the simplifying assumption that the attributes are independent. It is possible to more accurately estimate the joint probability distribution of the attributes; but this would be time consuming and require a lot of data.In each iteration, the artificially generated examples are labeled based on the current ensemble. Given an example, we first find the class membership probabilities predicted by the ensemble. We replace zero probabilities with a small non-zero value and normalize the

probabilities to make it a distribution. Labels are then selected, such that the probability of selection is inversely proportional to the current ensemble's predictions.

The experiments varies the amount of artificially generated data, Rsize; and found that the results do not vary much for the range 0.5 to 1. However, Rsize values lower than 0.5 do adversely affect DECORATE, because there is insufficient artificial data to give rise to high diversity. The results are for *Rsize* set to 1, i.e. the number of artificially generated examples is equal to the training set size

4.3 Advantage of Decorate method

Ensembles of classifiers are often more accurate than its component classifiers if the errors made by the ensemble members are uncorrelated. By training classifiers on oppositely labeled artificial examples, DECORATE reduces the correlation between ensemble members. Furthermore, the algorithm ensures that the training error of the ensemble is always less than or equal to the error of the base classifier; which usually results in a reduction of generalization error.

On average, combining the predictions of DECORATE ensembles will improve on the accuracy of the base classifier.

We believe that diversity is the key to constructing good ensembles, and is thus the basis of our approach. Other ensemble methods also encourage diversity, but in different ways. Bagging implicitly creates ensemble diversity, by training classifiers on different subsets of the data.
DECORATE will outperform Bagging and AdaBoost low on the learning curve i.e. when training sets are small.

V. Diversity Measurement

Homogeneous ensemble classifiers i.e. collections of base classifiers of a single model type. While heterogeneous ensemble classifiers, where the classifiers in the collection are not of the same type. There are two types of measures that have been used to study the diversity of ensembles: pairwise and non-pairwise.[7] Pairwise measures are designed to compare the differences in predictions of two classifiers.Their interpretation in that setting is clear, but once averaged over all possible pairs in a base classifier set, the interpretation may become less clear. Non-pairwise diversity measures are designed to measure differences in predictions of sets of more than two classifiers. Although their defenitions are typically more complex than the pairwise diversity measures. To generalize the pairwise diversity measures to an entire ensemble, we took the the average of the measurents over every pair of base classifiers. For a set of base classifiers B the average pairwise diversity can be calculated using:

$$Average = \frac{2\left(\sum_{i=1}^{|\mathcal{B}-1|}\sum_{j=i+1}^{|\mathcal{B}|} diversity_{i,j}\right)}{(n)(n-1)}$$

5.1 Disagreement Disagreement between a pair of classifiers, f and g, is the proportion of instances for which they predict different class labels. The range of this measure is between 0 (always agree),and 1 (always disagree).

$$D = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}_{f,g}(\mathbf{x}_i)$$

5.2 Double Fault Measure The double fault measure between a pair of classifiers is the proportion of instances for which they both predict the wrong class. The value of this measure is 1 when both of the classifiers are always wrong and 0 when the classifiers are never simultaneously wrong about the same instance.

$$DF = \frac{1}{n}\sum_{i=1}^{n}(1-\mathcal{O}_f(\mathbf{x}_i))(1-\mathcal{O}_g(\mathbf{x}_i))$$

where

$$\mathcal{O}_f(\mathbf{x}_i) = \begin{cases} 1 : f(\mathbf{x}_i) = y_i \\ 0 : f(\mathbf{x}) \neq y_i \end{cases}$$

5.3 Coincident Failure Diversity Coincident failure diversity (CDF) is a measure whose value is highest (1), when misclassifications are unique to one base classifier and lowest (0) when all base classifiers always make

the same class label predictions. Let pi denote the probability that exactly i of the L base classifiers predict the wrong class label for a randomly chosen instance. Then coincident failure diversity is defined as follows:

$$CFD = \begin{cases} 0 & : p_0 = 1 \\ \frac{1}{1-p_0} \sum_{i=1}^{L} \frac{L-i}{L-1} p_i & : p_0 < 1 \end{cases}$$

VI. Conclusion

If modification is there in artificial training examples, DECORATE is able to use a strong base learner to provide an accurate, diverse ensemble output. This method provide highly accurate ensembles that outperform both Bagging and Boosting low on the learning. Moreover, given large training sets, DECORATE outperforms Bagging and is competitive with Boosting.

References

[1] Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation and active learning.(Krogh & Vedelsby, 1995).

[2] J. Basilico, D. Dunlavy, S. Verzi, T. Bauer, and W. Shaneyfelt. Yucca mountain LSN archive assistant. Technical Report SAND2008-1622, Sandia National Laboratories, 2008.

[3] Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging.In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), pp. 1–10.

[4] Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In Proceedings of the Eighth ACM SIGKDD International Conferenceon Knowledge Discovery and Data Mining (KDD-2002), Edmonton, Alberta.

[5] Kuncheva, L., & Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with ensemble accuracy. Machine Learning, 51(2), 181–207.

[6] Melville, P., & Mooney, R. J. (2003). Constructing diverse classifier ensembles using artificial training examples. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003), pp. 505–510, Acapulco, Mexico.

[7] R. Ban_eld, L. Hall, K. Bowyer, and W. P. Kegelmeyer.A comparison of decision tree ensemble creation techniques. IEEE Trans. Pat. Recog. Mach. Int., 29(1):173{180, 2007.