Multiclass Learning by Boosting Bootstrap LDA Projections

Tuo Wang
Computer Sciences Department, University of Wisconsin - Madison
tuowang@cs.wisc.edu

Abstract

In the final project, I implement a multiclass classification algorithm based on algorithm Adaboost.M2. Firstly, a large number of bootstrap training subsets are sampled from the original training set and Fisher Linear Discriminant Analysis (LDA) is implemented in each subset to get a large number of LDA projections. Then at each iteration of Adaboost.M2, the projection with the minimum weighted k Nearest Neighbor (kNN) classification error is selected from a pool of linear projections to generate the final strong classifier. Finally, experiment results comparing with traditional LDA and bagging LDA are shown.

Introduction

As for classification problem, Principal Component Analysis (PCA) (I. T. Jolliffe 2002) may be the most popular methods in data representation and dimension reduction, but PCA does not take into account any difference in class. Fisher Linear Discriminant Analysis (LDA) (R. A. Fisher 1936) explicitly attempts to model the difference between the classes of data. LDA tends to find a set of projection vectors W maximizing the ratio of determinant of S_b to S_w: $W = \arg\max_w \frac{W^T S_b W}{W^T S_w W}$. S_b and S_w are defined as $S_w = \sum_{i=1}^{L} \sum_{\bar{x}_k \in X_i} (\bar{x}_k - \bar{m}_i)(\bar{x}_k - \bar{m}_i)^T$, $S_b = \sum_{i=1}^{T} n_i (\bar{m}_i - \bar{m})(\bar{m}_i - \bar{m})^T$ where L is the class number, each class X_i has n_i samples, $\bar{m}$ is the center of the whole training set, $\bar{m}_i$ is the center for the class X_i, and $\bar{x}_k$ is the sample belonging to class X_i . LDA has its limitations (D. Masip, J. Vitrià 2006): Gaussian assumption over the class distribution of the data samples, S_w may be singular, and the dimensionality of the subspaces obtained is limited by the number of classes. To improve the classification performance, many modification of LDA is proposed, such as Nonparametric Discriminant Analysis (NDA) (K. Fukunaga, J. Mantock 1983) and Regularized Discriminant Analysis (RDA) (J. H. Friedman 1989).

Algorithm Adaboost.M2 (Y. Freund, R. E. Schapire 1997), focusing on multiclass classification problem, is a transmutation of Adaboost algorithm. In AdaBoost.M2, each weak classifier has to minimize the pseudo-loss instead of the error rate. As long as the pseudo-loss is less than 1/2, which is easily reachable for weak base classifiers, an exponential decrease of an upper bound on the training error rate is guaranteed (Gunther Eibl etc, 2003).

Boosting Bootstrap LDA Projections

Assume that there are C classes, N samples in training set and D is the dimensionality of original samples, the implemented algorithm firstly samples a large number of bootstrap training subsets from the original training set and implements LDA on each subset to get a large number of bootstrap LDA projecting directions. Then at each step of Adaboost.M2, the projection with minimum weighted kNN classification error is selected to construct the final strong classifier. The algorithm has the following three steps.

Bootstrap Sampling

In order to create a bootstrap training subset of size N (shown in the following table) I perform N multinomial trials where, in each trial, one of the N samples is drawn into the subset. Some samples could be represented in the new subset once, twice or even more times and some samples may not be represented at all. By this mean, every bootstrap training subset is different from each other, and I create M such bootstrap training subsets.

```
Do for m=1, 2, …, M
    1. $S_m$ = Ø
    2. Do for i=1, 2, …, N
        ● z = random_integer_from {1, 2, …, N}
        ● Add xz to $S_m$
    3. Return $S_m$
```

Computing Bootstrap LDA Projections

LDA is then implemented for each of the M bootstrap training subsets to calculate a D×(C-1) projection matrix. RDA is used if S_w is singular. Hence, if we project the original data in to the new C-1 dimension space, we will get a new distribution which may be easier to be classified. They are the so called bootstrap LDA subspaces.

Boosting Bootstrap LDA Projections

The Adaboost.M2 algorithm is used to produce a single strong classifier that classify by voting the weighted predictions of a combination of weak learners which are generated or selected in a series of iterations.

CS 769 Class Project

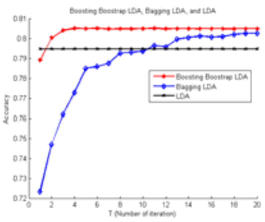
In Adaboost.M2 algorithm, a weak learner here is defined as a (C-1)-dimensional projection with a kNN classifier. For each sample, I use a D×(C-1) projection matrix to map a sample into (C-1)-dimensional subspace, and then classify it according to its k nearest neighbors. The output of this weak learner is defined as the proportion of every class' samples in the k nearest neighbors.

For each weak learner, training error is calculated based on the weighting function of each training sample to each class. In each of the iterations, the weak learner with the minimum training error is selected to generate the final classifier. The training error of the chosen weak learner is then used to determine its voting weight in the final combination and also to update the weighting function of all training samples. At last, classification result is made by the voting of these chosen weak learners with their voting weight. Following is a flow chart of the algorithm.

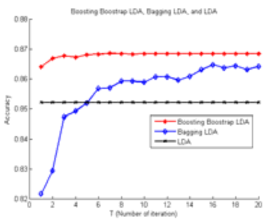
```
Input:
    N training samples $<(x_1,y_1),\dots,(x_N,y_N)>$ with labels $y \in Y = \{1,2,\dots,C\}$
    Integer M : the number of bootstrap training subsets.
    Integer T: the number of Adaboost.M2 training iterations.
Algorithm:
    1. Sample M bootstrap training subsets from the original training set, each subset has N samples.
    2. Perform LDA or RDA in each subset to get M bootstrap projections i.e. M WeakLearn classifiers.
    3. Boosting bootstrap projections by Adaboost.M2 algorithm:
        Initialize the distribution D over the N examples: D(i) = 1/N for i=1, …, N; the weight vector $w_{i,y}^1$ of the training samples: $w_{i,y}^1 = D(i)/(C-1)$ for i=1, …, N, $y \in Y - \{y_i\}$.
        Do for t = 1,2,…,T
            a) Set $W_i^t = \sum_{y \neq y_i} w_{i,y}^t$, $q_t(i,y) = \frac{w_{i,y}^t}{W_i^t}$, for $y \neq y_i$;
               and set $D_t(i) = \frac{W_i^t}{\sum_{i=1}^N W_i^t}$
            b) Call M WeakLearns, providing it with the distribution Dt and label weighting function qt; the pseudo-loss to each WeakLearn is computed by
               $\varepsilon_t = \frac{1}{2}\sum_{i=1}^N D_t(i)\left(1 - h_t(x_t,y_t) + \sum_{y \neq y_i} q_t(i,y) h_t(x_i,y_i)\right)$
               Then get back a hypothesis $h_t: X \times Y \to [0,1]$ with the minimum pseudo-loss $\varepsilon_t$.
            c) Set $\beta_t = \varepsilon_t/(1-\varepsilon_t)$.
            d) Set the new weights vector to be
               $w_{i,y}^{t+1} = w_{i,y}^t \beta_t^{\left(\frac{1}{2}\right)(1+h_t(x_i,y_i)-h_t(x_i,y))}$
               for i=1,…, N, $y \in Y - \{y_i\}$..
Output: The strong classifier
    $H(x) = arg \max_{y \in Y} \sum_{t=1}^T (\log \frac{1}{\beta_t})(h_t(x,y))$
```

Experimental Result

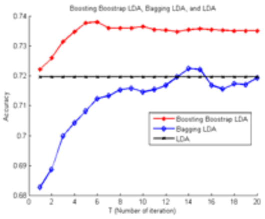
I perform experiment on four datasets: the MNIST hand written digits dataset (Y. LeCun) (10 classes), and three UCI machine learning repository datasets (A. Asuncion, D.J. Newman 2007): landset satellite dataset (6 classes), vehicle silhouettes dataset (4 classes), and splice-junction gene sequences dataset (3 classes). Except running my code on the original datasets, I also create some more complex datasets, e.g. in MNIST dataset, merge digits {1, 2, 3} into a new class I, {4, 5, 6} into another class II, and {7, 8, 9} into class III; for the landset satellite dataset, I merge class {1, 2} into a new class I, class {3, 4} into II, and class {5, 6} into III. To make comparison, traditional LDA and Bagging LDA are used.



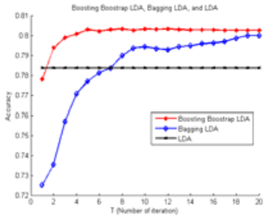
MNIST hand written digits (3 classes)



Landset satellite dataset (3 classes)



Vehicle silhouettes dataset (4 classes)



Gene sequences data (3 classes)

As for the above figures, horizontal axis is an index of iteration number, and vertical axis is classification accuracy on testing data. The first two figures are the results from datasets that are merged from the original set. The last two are based on the corresponding original dataset. For every class, I randomly pick up 200 samples from the original training dataset as my training data, and another 100 samples from the original testing dataset as my testing data. Then I run these algorithms 10 times, and compute the average classification accuracy. (Above results may not be very clear due to space limitation, so please download all the results at: http://pages.cs.wisc.edu/~tuowang /cs769/result.rar).

The experiment results demonstrate that the implemented method achieves better performance than other algorithms, especially when iteration *T* is small. The implemented algorithm reaches a nearly stable accuracy when *T*>6.

Conclusion

In this project I implemented a multiclass classification method based on algorithm Adaboost.M2. The advantage of the implemented algorithm is that it primarily performs by directly selecting the discriminant features with the minimum training error. Since each weak learner only represents a partial data distribution, the final boosted classifier may not be restricted to the global data distribution. The disadvantage is that this algorithm is very time consuming. Experiments on four real world datasets demonstrate the superiority of this method on classification accuracy comparing with LDA and Bagging LDA.

References

I. T. Jolliffe. 2002. Principle Component Analysis. Springer.

R. A. Fisher. 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7:179–188.

K. Fukunaga, J. Mantock. 1983. Nonparametric discriminant analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence 5:671–678.

J. H. Friedman. 1989. Regularized Discriminant Analysis. Journal of the American Statistical Association 84:165-175.

Y. Freund, R. E. Schapire. 1997. A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Sciences 55:119-139.

Y. LeCun. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/index.html.

A. Asuncion, D.J. Newman,. 2007. UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html.

D. Masip, J. Vitrià. 2006. Boosted discriminant projections for nearest neighbor classification," Pattern Recognition 39:164-170

Gunther Eibl, Karl–Peter Pfeiffer, 2003, Multiclass-Boosting for Weak Classifiers, Journal of Machine Learning Research.