Adaptive Learning Algorithm for Pattern Classification

Maohu Zhu, Nanfeng Jie and Tianzi Jiang
LIAMA Center for Computational Medicine, National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China
mhzhu@nlpr.ia.ac.cn

***Abstract*—In this paper, a pattern classification task was regarded as a sample selection problem where a sparse subset of sample from the labeled training set was chosen. We proposed an adaptive learning algorithm utilizing the least square function to address this problem. Using these selected samples, which we call informative vectors, a classifier capable of recognizing the test samples was established. This novel algorithm is a combination of searching strategies that, not only based on forward searching steps, but adaptively takes backward steps to correct the errors introduced by earlier forward steps. We experimentally demonstrated on face image and text dataset that classifier using such informative vectors outperformed other methods.**

Keywords—pattern classification; sample selection; informative vector; sparse representation; face recognition; text categorization

I. Introduction

Given a training set X of a number of samples {x_1, x_2, …, x_n} with known labels {y_1, y_2, …, y_n}, the goal of a classification algorithm is to infer a decision function $y = d(x)$ from the labeled training set. The decision function should predict the correct output value for any valid input object *x*. In order to measure the quality of the decision function, a loss function L(x) is defined. In the current paper, we limit ourselves to the least square function:

$$L(x,X) = \min_{w\in R^n} \|X * w - x\| \qquad (1)$$

$$= \min_{w_i\in w} \sum_{i=1}^{n} \|w_i{}^* x_i - x\|$$

Where w=[w_1,…,w_n], is the weight vector of all training samples in the sample space and *x is* a test sample. It is known that the least square often yields a poor generalization performance because the solution *w* overfits the data. To solve this issue, a small group of samples should be selected from the training samples to build a sparse decision function. A celebrated instantiation is in learning the prediction function of Support Vector Machine (SVM) [1], which only utilizes a limited subset of support vectors to characterize the decision boundary between two classes, rather than directly use all training examples. In practice, it is often difficult to infer a sparse decision function from training examples, since we may not be clever enough to find the sparse representation of model. L_0-norm regularization is a good learning method for sparse solution in learning a target function of model, which corresponds to the non-convex function:

$$L(x， X) = \underset{w_i}{Min} \sum_{i=1}^{n} \| w_i * x_i - x \| + \lambda \|\mathrm{w}\|_0 \qquad (2)$$

$$\| w \|_0 = | \{ i : w_i \neq 0 \} |$$

However, a fundamental difficulty with this method is the computational cost, because the number of subsets of {1, … ,n} of cardinality *k* (corresponding to the nonzero components of w) is exponential in *k*. It can be shown that the solution of this method is NP-hard[2]. In the current, there are no efficient algorithms to solve NP-hard problem. Due to computation difficulty, l_0-norm regularization is replaced by l_1-norm that is the closet convex approximation. It is known that l_1 regularization is of lead to sparse solutions. A promising technique called the lasso was proposed by Tibshirani[3] as follow:

$$L(x， X) = \underset{w_i}{Min} \sum_{i=1}^{n} \| w_i * x_i - x \| + \lambda \|\mathrm{w}\|_1 \qquad (3)$$

L_1 regularization is often applied to solving the problem of feature selection. John et al. employed l_1-regularizaion to selecting the relevant training samples for the recognition of face image[4]. In order to generate a sparse solution, a large regularization parameter is required. However, the l_1 penalty not only shrinks the irrelevant variable to zero, but shrink relevant variables to zero[5]. Instead, greedy search strategies are known by experimentalists to be computational advantageous and less prone to overfitting [6]. In this study, we proposed an adaptive learning algorithm to select a sparse subset of informative vectors that together recognize each test example.

II. Adaptive learning algorithm

The selected samples, called informative vectors henceforth, are used to establish a classifier. Based on square error, we proposed an adaptive learning algorithm that combines forward searching steps and backward adjusting steps. Unlike SVM, instead of choosing support vectors for all the test samples at once, a group of informative vectors for each test example is drawn in the current algorithm.

First, starting with the null model without any training example, the pattern x_i , for which $L(F \cup \{x_i\})$ is the smallest (i.e., x_i decreases squared error the greatest), is added to the current set F by forward steps to aggressively reduce the squared error at each step. This process keeps going on until the decrement of squared error falls below a given threshold ε (0.005 in this study). However, such procedure has a main shortcoming, that the selected subsets of samples is nested, where the subset F_k selected in step *k* is always included by the subset F_{k+1}. This implies that the errors caused in earlier forward steps would never have a chance to be removed. Consequently, backward elimination steps should be carried out to rectify these errors. The key design of this combination is to balance the forward and backward steps. The backward steps should not only fix the errors induced by earlier forward steps, but also keep as many achievements as possible. The pseudocode of the adaptive learning algorithm follows.

```
Input:      $X = [x_1, \dots, x_n] \in R^{m\times n}$ for k classess
            a test sample x
Initialize: the column of X was linearly scaled to [0,1]
            $S = [1, \dots, n], F = \emptyset, w = \emptyset, k = 0,$
            $\varepsilon = 0.005$ and $J_0 = \infty$
Output: F, w
while lengh(S) > 0
{
    $k = k + 1;$
    $[i_k, w_k, J_k] = \text{argmin}_{i\in S}||x, X(:, F \cup \{i\})||_2;$
    $\delta^+ = J_k - J_{k-1};$
    if ($\delta^+ < \varepsilon$)
    {
      $k = k - 1;$
      break;
    }
    $F = F \cup \{i_k\};$
    $S = S - \{i_k\};$
    $w = w_k;$
    while ($k > 1$)
    {
      $[j_k, w_k^-, J_k^-] = \text{argmin}_{j\in F}||x, X(:, F - \{j\})||_2$
      $\delta^- = J_k^- - J_k;$
      if ($\delta^- < 0.5 * \delta^+$)
      {
        $S = S \cup \{j_k\};$
        $F = F - \{j_k\};$
        $w = w_k^-;$
        $k = k - 1;$
      }else
        break;
      }
}
```

Note that backward steps were only carried out when the squared error increment δ^- is no more than half of the squared error decrement in the earlier corresponding forward step δ^+. This means that as long as n forward steps have been performed, no matter how many backward steps occur in this procedure, square error will decrease by at least $n\varepsilon/2$, which implies that the algorithm will automatically terminate after finite forward steps.

Together with informative vectors F and weight w, a sparse decision function *D*(*x*) can be derived. Thus, we define the decision function as follow:

$$d(x) = \arg\min_{i\in\{1,\dots,k\}} \| x, F_i * w_i \| \qquad (4)$$

Where F_i is the subset of informative vectors that belong to the *i*th class, and w_i corresponds to their weights, respectively. To classify a test sample x into a class, the decision function minimizes the residual between x and all informative vectors from this class.

III. Experiment Results

A number of classification experiments were implemented on three publicly available databases to estimate the efficacy of the proposed classification algorithm and meanwhile compare it with other machine learning algorithms. The three databases considered are: (1) Extended Yale B database, (2) CMU Face database, (3) Db world e-mails database (see TABLE I).

TABLE I
The Extended Yale B database consists of 2,414 frontal-face images of 38 individuals[7]. The cropped and normalized 192×168 face images were captured under various laboratory-controlled lighting conditions[8]. CMU Face and Db word e-mail datasets from the UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html)

Data set	Types	Instances	Features	Classes
Extended Yale B Database	Images	2,414	32,256	38
CMU Face	Images	640	3,840	20
Db world e-mails	Text	64	4,702	2

Separating data into training and testing sets is crucial for evaluating prediction models. Typically, when partitioning a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing. In order to avoid the possible bias introduced by relying on any one particular division into test and train components, a leave-one-out (LOO) cross-validation is used to split the *p* patterns into a training set of size *p*-1 and a test of size 1 and average the classification error on the left-out pattern over the *p* possible ways of obtaining such a partition. The advantage is that all the data can be used for training - none has to be held back in a separate test set. The results in TABLE II demonstrated that our method outperformed linear SVM and k-nearest-neighbor algorithm (KNN) on three different dataset.

TABLE II

Comparison of different learning algorithm: due to the high-dimensionality feature (32,256) of images from extended Yale B dataset, f-score feature selection was used to select top 3,000 discriminative features on this dataset for speeding the computation [9].

Methods	Extended Yale B Database	CMU Face	Db world
KNN	79.99%	99.19%	87.5%
*SVM (linear kernel)	95.23%	99.19%	57.8%
Our method	**99.01%**	**100%**	**89.06%**

* Nonlinear SVMs employ sophisticated kernel functions to classify data sets with complex decision surfaces. Determining the right parameters of such functions is not only computationally expensive, the resulting models are also susceptible to overfitting due to their large Vapnik Chervonenkis (VC) dimensions[10]. Instead of nonlinear kernel, the use of linear kernel makes it possible to directly compare between different algorithms working in the same feature space

Scaling them before dealing these dataset is very important. The main advantage is to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because the least square error usually depend on the inner products of feature vectors. In this study, each attribute of all dataset were linearly scaled to the range [0, 1].

IV. CONCLUSION

In this paper, pattern classification was deemed as a problem of selecting informative vectors from training samples. We proposed an adaptive learning algorithm to choose a sparse subset of informative vectors for classifying each test sample. Different from SVMs, the proposed algorithm is instance-based learning that, instead of performing explicit generalization, compare new problem instances with instances seen in training, which have been stored in memory. One advantage of this algorithm is that it has zero empirical risk and infinite VC dimension. Unlike KNN that require the orthogonality assumptions about samples, our algorithm utilizes mutual information between samples. We showed experimentally on three different databases that taking into account mutual information between samples in the informative vectors selection process impacts classification performance and yielded better classification than conventional learning algorithms. Beyond pattern classification, an intriguing question for future work is whether this model can be applied for object detection

.

ACKNOWLEDGMENT

This work was partially supported by the National Key Basic Research and Development Program (973) (Grant No. 2011CB707805), the Natural Science Foundation of China(Grant No. 91132301, No. 81101000 and No. 30900487), the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2011J098).

.

REFERENCES

[1] Cortes C, Vapnik V (1995) Support-Vector Networks. Machine Learning 20: 273-297.

[2] Amaldi E, Kann V (1998) On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theoretical Computer Science 209: 237-260.

[3] Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological): 267-288.

[4] Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31: 210-227.

[5] Zhang T (2011) Adaptive forward-backward greedy algorithm for learning sparse representations. Information Theory, IEEE Transactions on 57: 4689-4708.

[6] Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine learning 46: 389-422.

[7] Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. Pattern Analysis and Machine Intelligence, IEEE Transactions on 23: 643-660.

[8] Lee K-C, Ho J, Kriegman DJ (2005) Acquiring linear subspaces for face recognition under variable lighting. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27: 684-698.

[9] Chen Y-W, Lin C-J (2006) Combining SVMs with various feature selection strategies. Feature Extraction: Springer. pp. 315-324.

[10] Cheng H, Tan P-N, Jin R. Localized Support Vector Machine and Its Efficient Algorithm; 2007. Citeseer.