Feature Selection in Distance Learning from Small Sample

Lior Zamir

July 17, 2007

Abstract

Learning from a small sample is an acute problem which arises in many applications where acquiring new samples is difficult, time consuming or expensive. The problem becomes even harder when dealing with rich high dimensional data. The learning process in such cases is often preceded by dimensionality reduction or feature selection. The need to avoid overfitting of an algorithm to the data is critical in this situation and is sometimes handled using regularization techniques. In this work we address these problems by incorporating a feature selection method into an existing boosting based distance learning algorithm (DistBoost). Given equivalence constraints over pairs of data points, the feature selection method optimizes an L1 based cost function which takes into account the constraints and a regularization term. Selection is performed in each boosting iteration, with the training data weights causing the features relevant to the current data distribution to be chosen. We tested our algorithm on the recognition of facial images, using two public domain databases. We show the results of extensive experiments where our method outperforms a number of competing methods including the original boosting based distance learning method and two commonly used Mahalanobis distance functions.

Contents

1 Introduction

In the field of computational learning theory, algorithms are designed which attempt to learn certain properties of a given data set. The question of which properties are relevant is task dependent and can vary even when looking at the exact same data. As an example we can look at the task of classifying facial images. A set of facial images can be classified according to the different individuals appearing in the image, according to the gender of the individuals, their age or the facial expression. Each such task defines which are the important characteristics of the data. In the supervised (or semi-supervised) learning scenario, a simple way of guiding an algorithm towards these characteristics is by labeling the data. The labels can be presented in the form of the class attribution of a data sample or, as in this work, it can be in the form of equivalence constraints stating, for pairs of samples, whether they belong to the same class or not.
When asking how many examples we need to see in order to infer a general property or classification rule, a number of considerations are involved. While this is generally an open question, in a statistical framework, under certain assumptions, the larger the sample size is, the more accurate our modeling of the data would be. On the other hand practical issues, such as the time and cost needed to acquire the data and the time needed for the algorithm to learn, make it a desirable property to learn from as few examples as possible. The problem of learning from small sample has received much attention and approaches for dealing with it often require the use of additional bias information. This can be in the form of unlabeled data or labeled data from related classes which share a similar structure ([4],[2],[1],[8],[6]).
Another factor which complicates the problem further is the fact that the representation of the data may include irrelevant or misleading information (images of objects taken under different illumination conditions may cause different objects look similar and vice versa).
The effect of these obstacles can be greatly reduced by using feature selection methods, allowing the algorithm to concentrate on the relevant information. In this work it is done by minimizing a cost function which takes into account the equivalence constraints and a regularization term, whose purpose is twofold. Regularization techniques are used to prevent overfitting parameters to the given data, and this is especially important when dealing with small data. In the form presented here, the regularization term often constrains the problem to sparse solutions, effectively causing the selection of relevant features.

1.1 Learning Distance Functions

Distance functions are mappings from pairs of data points to the real numbers. Intuitively, one would like a distance function which maps data points according to a (inverse) similarity notion that humans use, for example when comparing images. In this context we do not restrict ourselves to metric distances (conforming to the properties of isolation, symmetry and the triangle inequality) which have been shown not to reflect human similarity judgements ([9]).
Some of the motivations for learning distance function and not mearly classification rules include:

- Many clustering and classification methods receive as their input the dis-

tances between pairs of data points. A good distance measure would improve their performance.

- Data is not necessarily divided into clear classes. Often there is a soft transition between one class to another and binding the description capability to just class attribution is quite limiting. For example in the application of image retrieval, say we have learnt to distinguish between images of land mammals and birds but we are given a query of a platypus - both mammals and birds seem like good candidates for retrieval.

- Another motivation related to the above metioned is the "Learning to Learn" scenario. Inspired by human's ability to transfer knowledge about a certain task into new similar tasks, it is suggested that this can be done with the help of distance functions ([8],[5]).

Defining what a good distance function is, is somewhat problematic because of the vagueness of the target concept, yet some sort of assesment of the quality of the learnt function still needs to be given. Several methods are suggested and used in this work and are described in 4.2.
Various algorithms have been developed for learning distance functions, of which some are used as comparison to the method proposed here (see 4.1 for a description of these methods). One common family of distance functions is the Mahalanobis distance, which is characterized by a positive semi-definite matrix A with the distance between row data vectors x, y defined as $(x-y)^T \cdot A \cdot (x-y)$. When A is the identity matrix this is the Euclidean distance.
Another important family of distance functions used widely for classification tasks are kernel functions. In methods such as SVM, these functions are used as computationaly efficient ways of calculating the dot product in high dimensional spaces, where data is hopefully more easily separated. The importance of carefully selecting the appropriate kernel for the given task has lead to the developement of kernel learning algorithms, one of which is based on the distance learning algorithm used in this work ([4]).
Our feature selection method uses a powerful non parametric distance learning algorithm called DistBoost([3]). The algorithm boosts weak distance functions and is described in the following sections.
For a thorough review on distance function learning see [5].

1.2 Boosting

Boosting is a machine learning technique which uses a set of weak hypotheses and amplifies their perfomance by combining them and creating a stronger hypothesis. Many variants of this idea have been developed in recent years due to theoretical justifications as well as practical successes of the algorithms. The classical AdaBoost algorithm was developed by Freund and Schapire (1997) and a generalization of it was presented by Schapire and Singer(1999) which allows the usage of confidence rated predictors as weak hypotheses([7]).
AdaBoost takes a training set as input and proceeds in iterations. Each iteration a weak hypothesis is greedily chosen with the aim of minimizing a cost function defined over the labeled data. A distribution over the training set is maintained and the weights are updated so as to give higher weights to samples

which were misclassified using the chosen hypothesis. The final strong hypothesis is a weighted sum of the chosen hypotheses.
AdaBoost tends to avoid overfitting, which makes it a good candidate for our purpose of learning from small samples. The next section describes an application of Adaboost used for distance learning with an extension for use with unlabeled data - the DistBoost algorithm.

1.3 DistBoost

DistBoost([3]) is a semi-supervised algorithm for learning distance functions. Its input is a training set augmented by partial side information in the form of equivalence constraints. The algorithm is based on boosting Gaussian mixture models which are computed using constrained EM. It therefore extends Adaboost to the case where part of the data is unlabeled. DistBoost has been shown to significantly outperform other distance learning methods when used for clustering and nearest neighbor classification. A variant of this method is also used in kernel function learning (KernelBoost [4]) where good performance is achieved on small training samples. Following is a schematic description of the algortihm:

Algorithm 1 DistBoost - Boosting with unlabeled data

Given $(x_1, y_1), ..., (x_n, y_n)$; $x_i \in X$, $y_i \in \{-1, 1, *\}$ Initialize $D_1(i) = 1/n \;\; i = 1, .., n$
For $t = 1, .., T$

1. Train weak learner using distribution D_t
2. Get weak hypothesis $h_t : X \to [-1, 1]$ with $r_t = \sum_{i=1}^{n} D_t(i)h_t(i) > 0$. If no such hypothesis can be found, terminate the loop and set $T = t$.
3. Choose $\alpha_t = \frac{1}{2} \ln(\frac{1+r}{1-r})$
4. Update:

$$D_{t+1}(i) = \begin{cases} D_t(i) \exp(-\alpha_t y_i h_t(x_i)) & y_i \in \{-1, 1\} \\ D_t(i) \exp(-\alpha_t) & y_i = * \end{cases}$$

5. Normalize: $D_{t+1}(i) = D_{t+1}(i)/Z_{t+1}$
   where $Z_{t+1} = \sum_{i=1}^{n} D_{t+1}(i)$
6. Output the final hypothesis $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

A drawback of this method when used on small, high dimensional samples is that fitting Gaussian mixture models in the original representation of this data requires the estimation of many parameters (often more parameters than data samples), which makes the weak hypotheses highly inaccurate (and also very costly in computation time). This is traditionally overcome by reducing the dimension of the data using PCA or LDA, but as results show, this reduction

may cause the loss of valuable information. In this work we suggest handling this problem using feature selection.

2 Feature Selection

2.1 L1 Based Feature selection

3 Algorithm

3.1 Feature Selection

3.2 Incorporation into DistBoost

4 Experimental Results

4.1 Comparison with Other Distance Functions

4.2 Evaluation methods

4.3 Data Sets

4.4 Results

5 References

References

[1] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28:7–39, 1997.

[2] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[3] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *ICML*, 2004.

[4] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *ICML*, 2006.

[5] Tomer Herz. Learning Distance Functions: Algorithms and Applications. 2006.

[6] I.Martin L.Wolf. Robust boosting for learning from few examples. *IEEE Conf. on Computer Vision and Pattern Recognition(CVPR)*, 2005.

[7] Robert E. Schapire and Yoram Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[8] S. Thrun and L.Y. Pratt. *Learning To Learn*. Kluwer Academic Publishers, Boston, MA, 1998.

[9] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.