ISCA Archive
http://www.isca-speech.org/archive

International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)
Taipei, Taiwan
August 23-24, 2002

INCORPORATING PROBABILITY INTO SUPPORT VECTOR MACHINE FOR SPEAKER RECOGNITION

Tieyan FU, Qixiu HU, Guangyou XU

Department of Computer Science and Technology,
Tsinghua University, Beijing 100084
futieyan00@mails.tsinghua.edu.cn

xxs-dau@tsinghua.edu.cn

ABSTRACT

Support Vector Machines (SVMs) is basically a discriminative classifiers, while it is hopefully that incorporating probability into SVMs will achieve better performance. This paper briefly reviews some of the methods that can be used to carry out the combination. By following one of them, we make it suitable for the task of speaker recognition, and Gaussian Mixture Models (GMM) is used as the generative model to derive Fisher kernel. Preliminary experiments are performed on a speaker identification task. The results are compared with GMM and standard SVMs baseline systems, and some suggestions have been made for future direction.

1. INTRODUCTION

Generative probability models such as Gaussian Mixture Models (GMM) exhibits inherent ability of treating missing information in sequences analysis and the task of pattern classification is done through Bayes rule. The classification results are provided with a measure of confidence. Alternatively, discriminative model enable us to directly characterize the decision boundaries without the need to firstly assume some form of density function for sample data, which is what we do in generative methods, and it has been proven that discriminative model is superior to generative one in the context of classification. We hope that combining these two methods will not only preserve the discriminability but also provide confidence measure for classification task.

The Support Vector Machine is a new and very promising classification technique. The main idea behind the techniques is to separate the classes with a surface that maximizes the margin between them. In this paper, we use it as our discriminative model, and combine it with probability models. There have been several methods exist for such combinations [4, 5, 6, 7, and 8] which we generally categorize into three classes: (1) Probabilistic Interpretation; (2) Moderated Output; (3) Moderated Kernel. We follow the third method and derive the Fisher kernel from GMM, and test it in a speaker identification task.

The task of text-independent speaker identification is to identify a speaker from among a pool of candidates based on acoustic utterances, which are preprocessed into a sequence of acoustic data vectors. One widely used approach is to employ GMM to represent the distributions of observations obtained from a speaker, and the decision is made through choosing the speaker with maximum probability [1] [2]. In this paper we use it and standard SVMs as our baseline system.

The outline of this paper is as follows. In Section 2 we give an overview of methods that can be used to incorporating probability into SVMs. In Section 3 we describe the Fisher kernel and derive the formula of Fisher Kernel from the Gaussian Mixture Model. In Section 4 we present our experimental results. Finally, in Section 5 we give conclusions and suggestions for future work.

2. OVERVIEW OF COMBING SVM WITH PROBABILITY

2.1 Brief description of SVMs

Support Vector Machines [9] is a recent technique that has been widely applied to pattern classification and regression estimation. For convenience of later discussion, we give brief description of SVMs; see, e.g., [9, 10], for tutorials and overviews of recent developments.

In the case of two-class problem, we are given training data $\{x_i, y_i\}, i = 1,\ldots,l, y_i \in \{-1,1\}, x_i \in \Re^d$. Suppose we have some hyperplane which separate the positive examples from the negative ones. The point x lies on the hyperplane satisfy $w \cdot x + b = 0$. For the linear separable case, Support Vectors (SVs) are those points separating hyperplane with largest margin, where margin is defined as the sum of $d_+ + d_-$ (d_+ is the shortest distance between positive example and the separating hyperplane, d_- is defined similarly). The problem can be formulated as follow:

$$\text{Minimize } \frac{1}{2}\|w\|^2$$

$$\text{Subject to } y_i(x_i \cdot w + b) - 1 \geq 0, \forall i \qquad (1)$$

If the problem is not linearly separable, *slack variables* $\xi_i \geq 0$ for all i are introduced, and a penalty term $C\sum_i \xi_i$ has been added to the objection function with a penalty coefficient C.

2.2 Three methods for combination

In the development of SVMs, various schemes have been proposed to incorporating generative model into the framework of SVMs. We next review these methods by putting them into three categories.

2.2.1 Probabilistic Interpretation

This method interprets Support Vector Machines as maximum a posterior (MAP) solutions to inference problems with Gaussian Process priors [5]. It shows that the SVM kernel defines a prior over functions on the input space, avoiding the need to think in terms of high-dimensional feature spaces. Following equation (1), we get the optimization problem with slack variables:

$$\frac{1}{2}\|w\|^2 + C\sum_i l(y_i[w \cdot x_i + b]) \qquad (2)$$

Where $l(z) = (1-z)H(1-z)$ is the *hinge loss* function. H (a) = 1 if $a \geq 0$ and 0 otherwise. To interpret SVMs probabilistically, one can regard equation (2) as defining a negative log-posterior probability for parameter **w** and b of the SVMs. Assuming the components of **w** is uncorrelated with each other and have unit variance (normalized **w**), the first term gives a Gaussian prior on **w**,

$$Q(w) \sim \exp\left(-\frac{1}{2}\|w\|^2\right) \qquad (3)$$

[5] chooses a prior on b with variance B^2, which gives

$$Q(b) \sim \exp\left(-\frac{1}{2}b^2 B^{-2}\right) \qquad (4)$$

and obtains equation (2) by letting $B \to \infty$. The second term of (2) is data-dependent, and can be interpret as the likelihood of the training data given model parameters θ, i.e.,

$$Q(y = \pm 1 \mid x, \theta) = \exp(-Cl(y\theta(x)))/\lambda \quad (5)$$

where λ is a normalizing factor ensuring that the probability sum to one. Several authors [5, 6] have pointed out the relationship between SVMs and Gaussian Process (GP).

2.2.2 Moderated Output

This method generally trains SVMs using standard technique, which takes advantage of well-studied training algorithm and maintains the sparseness property of kernel machine. Then different probability to be estimated leads to two ways of modification. One way is to estimate *class-conditional densities* $p(f \mid y = \pm 1)$ [11]. It is assumed that the form of the densities function is Gaussian distribution, and the posterior probability is obtained through Bayes rule. However it is suspected that the assumption may not be in accord with the true probability. Another way is to fit the posterior probability $p(y = \pm 1 \mid f)$ directly [8]. The parametric form of the model is adapted to give the best probability outputs. The question again is how to choose the form of parametric model. By observing empirical data, Platt [8] suggests the following sigmoid function

$$p(y = 1 \mid f) = \frac{1}{1 + \exp(Af + B)},$$

where A and B are model parameters.

From a Bayesian perspective, [7] extends the use of moderated outputs by stating the relationship between evidence framework and the SVMs: training of the SVMs can be regarded as approximately performing the first level of inference in the evidence framework. The moderated output derived serves naturally as an approximation to the posterior class probability.

2.2.3 Moderated Kernel

The key idea of this method is to derive kernel function from a generative probability model. As kernel function implicitly describes metric of distance between examples, this metric may be defined through a generative probability function $p(x \mid \theta)$. In the context of classification, it is desired to capture the difference in generative process between a pair of examples, rather than simply the difference in the posterior probabilities for the label estimated for each of the examples, which is generally used for discrimination in simple generative approach. One way to defines the difference is in the gradient space of log probability the generative model, i.e.,

$$U_x = \nabla_\theta \log P(X \mid \theta) \qquad (6)$$

which maps examples into a fixed score space. This mapping is called *Fisher Score*, and the kernel $K(x_i, x_j) = U_{x_i}^T U_{x_j}$ is named *Fisher Kernel*. As our work follows this method, we will give details in the following section.

3. FISHER KERNEL FROM GMM

To derive a kernel function from a generative model, we need first to select one. In [4] the authors choose a logistic function for the task of DNA and protein sequence analysis. Since GMM and its variations have show state of the art performance [1, 2 and 3] in speaker recognition comparing with other methods, we prefer to use it to derive the Fisher Kernel. We also use it as baseline system in our experiments.

3.1 Generative Model Description

For text-independent speaker recognition, as there is no prior knowledge of what the speaker says, the most successful method is to represent speaker model by Gaussian mixture model. A GMM is a weighted sum of M component densities，we denote it as

$$p(x \mid \theta) = \sum_{i=1}^{M} w_i b_i(x) \qquad (7)$$

Where $\sum_{i=1}^{M} w_i = 1$, and $b_i(x)$ is a multivariate Gaussian function

$$b_i(x) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right\}$$

μ_i, Σ_i are mean and covariance matrix for the i^th component respectively. There are two steps in training specific speaker model. Given a collection of training samples from various speakers, we first train a Universal Background Model (UBM) using maximum likelihood criteria. The model is trained by pooling all of the available data together. Using the UBM as the initial model, we then adapt specific speaker model according to the rule of maximum a posteriori (MAP) with corresponding speech. The detail of training procedure can be found in [3]. As the UBM model is trained using various speeches, such as different gender, we hope it would include those alternative speech encountered during recognition. In comparing with other speaker-specific background schemes, this method also provides a nature way of score normalization. Next we will describe how to derive the Fisher Kernel from GMM.

3.2 Fisher Kernel

Given observation vectors $X = \{x_1, x_2, ..., x_T\}$ and a generative model $p(X \mid \theta)$, the Fisher Score is defined as

$$\varphi_\theta(X) = \nabla_\theta \ln p(X \mid \theta) \tag{8}$$

Here T is the length of the observation sequence. It is the first order derivative of the generative model $\ln p(X \mid \theta)$ with respect to model parameter θ. With this score-map, the kernel can be obtained as follow. For two vector observations sequences X_i, X_j , we get the Fisher Kernel

$$K(X_i, X_j) = \varphi_\theta(X_i)\Sigma_c^{-1}\varphi_\theta(X_j) \tag{9}$$

where Σ is the covariance matrix of the scores in the score-space,

$$\begin{aligned} \Sigma_c &= \sum_X (\varphi(X)-\mu_c)(\varphi(X)-\mu_c)^T p(X \mid \theta) \\ \mu_c &= \sum_X \varphi(X) p(X \mid \theta) \end{aligned} \tag{10}$$

With GMM as our generative model, the score-map for single vector x_t is define as

$$\begin{aligned} \nabla_\theta \ln p(x_t \mid \theta) &= \left[\nabla_{\theta_1} \ln p(x_t \mid \theta), ..., \nabla_{\theta_M} \ln p(x_t \mid \theta)\right]^T \\ \nabla_{\theta_i} \ln p(x_t \mid \theta) &= \begin{bmatrix} \nabla_{\mu_i} \ln p(x_t \mid \theta) \\ \nabla_{\Sigma_i} \ln p(x_t \mid \theta) \\ \nabla_{w_i} \ln p(x_t \mid \theta) \end{bmatrix}, i = 1, ..., M \end{aligned} \tag{11}$$

It is assumed that dimension of the observation vector is D and we use diagonal covariance matrix in GMM, then the dimension for each component is 2D+1, and the dimension of the score-map is (2D+1) * M -1. The constraint $\sum_{i=1}^{M} w_i = 1$ indicates that there are only (D-1) free weight parameters. Here we only show how to derive $\nabla_{\mu_i} \ln p(x_i \mid \theta)$, others are similar. With (7) we have, $\nabla_{\mu_i} \ln p(x_t \mid \theta) = \frac{1}{p(x_t \mid \theta)} \nabla_{\mu_i} p(x_t \mid \theta)$, and

$$\nabla_{\mu_i} p(x_t \mid \theta) = \frac{w_k}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} \nabla_{\mu_i} \exp\left\{-\frac{1}{2}(x_t-\mu_i)^T \Sigma_i^{-1}(x_t-\mu_i)\right\}$$

$$= \frac{w_i}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(x_t-\mu_i)^T \Sigma_i^{-1}(x_t-\mu_i)\right\}(x_t-\mu_i)^T \Sigma_i^{-1}$$

$$= w_i \mathrm{N}(\mu_k, \Sigma_k)(x_t - \mu_i)^T \Sigma_i^{-1}$$

Therefore,

$$\nabla_{\mu_i} \ln p(x_t \mid \theta) = \frac{w_i N(\mu_i, \Sigma_i)}{p(x_t \mid \theta)}(x_t-\mu_i)^T \Sigma_i^{-1}. \tag{12}$$

Another question specific to our problem is that the length of feature sequence is not fixed. We need to find a way to conveniently deal with the situation. A simple scheme has been employed. When the length of observation vectors is variable, we normalize this score-map by dividing it with the length T, i.e.,

$$\nabla_{\theta_i} \ln p(X \mid \theta) = \frac{1}{T}\sum_{t=1}^{T} \nabla_{\theta_i} \ln p(x_t \mid \theta)$$

4. EXPERIMENTS AND RESULT

In the task of speaker recognition, GMM and its variations [1, 2 and 3] have shown state of the art performance comparing with other methods. The baseline system we used in this paper follows the UBM-GMM scheme described in [3]. Standard SVMs has also been tested against our proposed system.

There are 200 speakers in our dataset. Each speaker has 60 seconds speeches for training and recognition respectively. Speeches from 40 speakers (20 males and 20 females) are pooled together to train the UBM while all speakers are tested. So the task can be considered as open set. The speaker-dependent GMM is trained using 60 seconds speech from each speaker. Speeches selected for training UBM are also used in the speaker-specific training phase for the same speaker. The 60 seconds testing speech are equally divided into eight parts with 7.5 seconds each, so there are eight test segments for each speaker. The front end processing consists of two main steps: feature vector extraction and speech activity detection. Feature vectors are composed of 16 mel-cepstra and 16 delta cepstra. These vectors are computed every 10 ms by windowing the input speech with a 20 ms Hamming window, computing the log magnitude FFT, and processing that through a 24-filter mel-filter bank. Speech activity is detected using an energy-zero cross rate detector. The product of energy and zero-cross rate are calculated every 10 ms, and the threshold is set experimentally.

LIBSVM [12] is applied in these experiments and small changes have been added to adapt it to our task. The speeches used in training SVMs are the same as those used in training GMM. The experiments are mainly conducted with two groups: multi-classes and binary classes. We employ the one-vs-other scheme to extend SVMs for multi-classes problem.

The training and testing of the standard SVMs baseline system is the same as SVM+GMM with one exception. In order to obtain sparse SVMs, we need reduce the number of feature vectors, and we introduce such an intermediate step: after front end processing, we group those feature vectors into clusters using nearest neighbor algorithm, and the SVMs is trained using representatives of those clusters.

With binary classes problem, our schemes have show very promising result. Randomly selecting two speakers to construct a binary problem, we form 100 such pairs with 8 testing segments for each pair. Out of all pairs 84 testing pair are correct for all 8 segments, and the error rates for the rest are less than 6.3%. It is useful for situations such as knowing prior the number of speaker presented, and the schemes described above can be easily adapted to it by selecting two best speakers using a few frames in the beginning.

With such a large speaker set, the best result we achieve is 71.2% in multi-classes case. Comparing with the baseline system, the result is not satisfied. There are some reasons for low recognition rate. Firstly the kernel needs to be normalized in the score-space. The normalization requires the inverse of covariance matrix $\sum_c$ that can only be approximated using equation (10). Here we assume the independence of each dimension in the score-space, and thus use a form of diagonal matrix as an approximation. This ignores the dependence between different components in GMM as well as means and variances in each component, thus it only provides a rough approximation. The limited training data even makes the situation even worse. [4] use Fisher Information matrix as an substitution. However it require $\eta_c = 0$ which is often not the case. So a more effective method may be employed. A possible way is to resample the available data according some statistical process that generate this sequence, and use this process to generate more data. The main difficult in this method is how well the process can be estimated, and the research is under development. As we use one-vs-other scheme, data imbalance between positive and negative examples can be another problem. This can be partly testified by the binary cases. One possible way is to employ one-vs-one scheme, but as there are 200 speakers, the total number of models (C_{200}^2) for all speakers would be too large. There is a need for a better way of dealing with this situation.

GMM	95%
SVM	63.4%
GMM+SVM	71.2%

We also use only part of the fisher score, i.e., only derivative with mean and weight or variance and weight. Similar results have been obtained and omitted here.

5. CONCLUSIONS

In this paper, we review some of the methods available to combine SVMs with generative model, and give a Fisher Kernel by using the first order derivative of GMM. This method is test in the task of speaker recognition. The results in binary testing show that this method is especially suit for binary classes. Comparing with standard SVM, our system give better result. Future work will involve effectively estimate covariance matrix in score-space and apply it in our speaker segmentation task.

6. REFERENCES

[1] D.A. Reynolds, and Rose C., *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*, IEEE Trans. On Speech and Audio Processing, vol.3, 1995.

[2] U. Chaudhari, J. Navratil, S. Maes, and G. Ramaswamy, *Very large population text-independent speaker identification*, in Proc. ICASSP, 2001

[3] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models.* In Digital Signal Processing 10, 19-41. 2000.

[4] T. Jaakkola and Haussler D., *Exploiting generative models in discriminative classifiers*, in M.S. Kearns, S.A. Sola, and D.A. Cohn, editors, Advances in Neural Information Processing System 11 : Proceedings of the 1998, pp. 487-493.

[5] P. Sollich, *Probabilistic methods for support vector machines, In S.A. Solla, T.K. Leen, and K.R. Muller, editors, Advances in Neural Information Processing Systems 12, pp. 349-355.*

[6] M. Seeger, *Relationships between Gaussian Processes, Support Vector Machines and Smoothing Splines*, Technical Report of Institute for Adaptive and Neural Computation University of Edinburgh.

[7] J.T. Kwok. *Moderating the Outputs of Support Vector machine Classifiers.* IEEE Transactions on Neural Networks, 10 pp1018-1031, 1999

[8] J.C. Platt. *Probabilistic Outputs for Support Vector machines and Comparisons to Regularized Likelihood Methods*, in P.B. Alexander, J. Smola, Bernhard Scholkopf and Dale Schuurmans, editors, Advances in Large Margin Classifiers., 1999.

[9] C.J.C. Burges, *A Tutorial on Support Vector machines for Pattern Recognition*, Data Mining and Knowledge Discovery, vol.2, no.2, pp.1-47, 1998

[10] B. Scholkopf, C. Burges, and A.J. Smola. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[11] T. Hastie and R. Tibshirani. *Classification by pair coupling.* Technical report, Stanford University and University of Toronto, 1996.

[12] Chih-Chung Chang and Chih-Jen Lin, Implementation of SVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/