CHAPTER 8

FUZZY SVM FOR SPEAKER CLASSIFICATION

8.1 INTRODUCTION

This chapter describes the application of Fuzzy Support Vector Machine (FSVM) for classification of speakers using GMM-Super vectors. In addition to the basics underlying the FSVM, we explain the ability of FSVM to automatically identify the optimum classes with less number of support vectors. FSVM classifies the data based on the fuzzy membership function and accordingly it disqualifies some of the data as surely known class data. It works only on the data which are difficult to classify into a single class. Introduction of fuzzy theory in support vector machine yields better classification accuracy and requires less number of support vectors. Experiments were conducted on 2001 NIST Speaker Recognition Evaluation corpus. Performance of the proposed GMM-FSVM based speaker verification system is compared with the conventional GMM-UBM and GMM-SVM based systems. Experimental results indicate that the fuzzy SVM based speaker verification system with GMM super vector achieves better performance to GMM-SVM system.

Recently, the idea of stacking the mean vectors of GMM model to form a GMM mean super vector has become successful in speaker verification using Support vector Machine (Campbell et al 2006).
Support Vector Machine (SVM) is a two-class classifier based on the principles of structural risk minimization. SVMs perform a non linear mapping from an input space to an SVM expansion space. Linear classification techniques are then applied in this potentially high-dimensional space.

In the early 1990s, SVMs were first proposed by Vapnik (1998) as optimal margin classifier. In pattern recognition works (Burges 1998), SVM had been used for isolated handwritten digit recognition (Cortes & Vapnik 1995), object recognition (Blanz et al 1996) and speaker verification (Kong Aik Lee et al 2011). Then, in order to combine the advantage of SVM and the state of art technique GMM-UBM for speaker verification system, a new GMM-SVM system was proposed by Campbell et al (2006). In this approach, the GMM super vector is applied as the input for SVM. The experiments done by Campbell et al (2006) using SVM-GMM and NAP variability compensation with 20 female and 20 male speakers has achieved an error rate of 0.4% and average accuracy rate of 95.1% with 22 order MFCCs for the 2004 NIST speaker recognition evaluation corpus.

The main design component in an SVM is feature space. Since inner products induce distance metrics and vice versa, the basic goal in SVM kernel design is to find an appropriate metric in the SVM feature space relevant to the classification problem. A study on the use of MFCC and SVM for text dependent speaker verification is carried out by Shi-Huang Chen et al (2009). By using discrete events and their probabilities from speech signal to construct super vectors based on Bhattacharyya distance as input for SVM, Kong Aik Lee et al (2011) obtained an Equal Error Rate (EER) of 5.51% and a Decision Cost Function (DCF) of 2.69. The performance of SVM depends on the selection of Kernel functions used to compute distances among data.
points. Mostly used kernel functions are polynomial, linear and Gaussian functions. Since these functions do not use the advantage of inherent probability distributions of data, a deterministic kernel based on KL divergence was proposed by Pedro Moreno et al (2003). Another drawback of GMM-super vector-SVM approach in speaker verification found by Wai Mak & Wei Rao (2011) is imbalance between the numbers of speaker class utterances and imposter class utterances. They proposed a method of utterance partitioning with acoustic vector resampling to reduce the error rate due to data imbalance problem. Zhao Jian et al (2007) was able to obtain an EER of 4.92% and a DCF of 0.0251 by using GMM-SVM with a Nuisance Attribute Projection kernel.

8.2 MACHINE LEARNING BY RISK MINIMIZATION

Generally, for classification, we need an automatic machine which produces the class label as the output for the input test vector. This classifier requires some training vectors in order to identify the class of test input vector. It determines the closest training vector for the test vector and assigns the class label associated with the closest training vector. This method of classification is simple one only when fewer training data is available. But for larger training samples, the classifier performs the mapping between training and testing vectors. In the mapping process it requires some parameters to be adjusted to make the classifier fit the data. That means that the classifier learns the training data by reducing the cost function which is a function of error. The error is the difference between the output of the classifier and the desired target for the training vector. The classifier’s expected error is approximated by the empirical risk.
The number of parameters to be adjusted depends on the distribution of data which is unknown. However, the performance of the classifier relies on the number of parameters. Too few parameters do not allow the classifier to perform well and too many parameters make the classifier to over train. To prevent the classifier from overtraining, a complexity term is included in the risk minimization. Thus the minimization of empirical risk becomes the regularised one and that provides a balanced performance of the classifier on the training vectors and complexity.

Support Vector Machines (SVMs) represents a machine learning technique. The SVM learning method is based on the principle of structural risk minimization. SVM does not minimize the objective function based on training; instead it attempts to minimize one bound on the generalization error. SVMs have been widely used due to their high generalization ability over a wide range of applications and because they have better test performance than other traditional learning machines.

8.3 BINARY LINEAR SVM CLASSIFIER

The basic concepts of binary classification by a linear SVM may be stated as follows. Given an input space $X = \{x_1, x_2, ... x_N\}$, an output binary space $T = \{+1, -1\}$ and the N training patterns $(x_i, t_i) \in S \subseteq X \times T$ where $i = 1, 2, ... N$. If $X \subseteq \mathbb{R}^n$, then $x_i \mapsto x_i$. The main aim of the binary classification is to search for a linear hyper plane $g(x) = w^T z + b$ where $z = \phi(x)$ denote the corresponding feature space vector with a mapping $\phi$ from $\mathbb{R}^n$ to a feature space $\mathbb{Z}$. The decision function of a hyper plane classifier is of the form which will estimate '1' for any $x$. 
Generally, in SVM we are looking for the hyperplane defined by ‘w’ & ‘b’ which maximize the margin between every training pattern and the hyperplane (Burges 1998). If we consider a two dimensional vector space (which has two classes), the decision boundary is a straight line which separates the two classes exactly. But if there are two boundaries which separate the two classes, the best boundary is the line which is exactly halfway between the two classes. To determine the best boundary, two parallel boundaries which can separate the classes are selected. Now, the two boundaries are moved apart as possible without making an error. These boundaries become margins of the two classes. The best boundary is the central line between these two margins. Maximizing the region between the two margins and determining the best boundary through convex hulls and normal vectors are shown in Figure 8.1 (a) and (b). The parameters of the line at the centre of two margins are adjusted to have the output of the classifier as either +1 or -1. This is described in the Figure 8.1 (c).

![Diagram](image)

(a) The decision boundary that is in midway between the two margins of the two classes

Figure 8.1 (Continued)
(b) Obtaining best boundary through convex hulls and normal vector

(c) Adjustment of boundary between margins

Figure 8.1 Concept of SVM binary classifications (Figure is taken from Burges 1998)
This linear optimum hyperplane will be the best decision surface only if the training set (positive and negative training vectors in feature space) are linearly separable. But for non separable training set the optimization problem is modified with the slack variable and a penalty term. This primal training problem has a complicated constraint set and limitation on dimensionality of feature space. These two difficulties can be overcome by modifying the SVM training problem with a dual variable $\alpha$ as

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) = \frac{1}{2} \alpha^\top G \alpha - \mathbf{1} \alpha$$  \hspace{1cm} (8.1)$$

such that $0 \leq \alpha \leq \frac{C}{N}$; $\mathbf{t}^\top \alpha = 0$

where $G \in \mathbb{R}^{n \times n}$ is the Hessian,

$$G_{i,j} = t_i^t t_j K(x_i, x_j) \quad \forall i, j \in \mathbb{Z}_n$$

$K$ is the Kernel function which satisfies Mercer’s condition[3]. Thus the optimal hyperplane can be reformulated as

$$g(x) = w^\top z + b = \sum_{i \in \mathbb{Z}_n} \alpha_i K(x_i, x_j) + b$$  \hspace{1cm} (8.2)$$

where $b = t_j - \sum_{i \in \mathbb{Z}_n} \alpha_i K(x_i, x_j) \quad \forall j \leq \alpha_j \leq \frac{C}{N}$. The dual variable $\alpha$ is associated with a training pair $(x_i, t_i) \in \mathcal{X}$.

8.4 GMM SUPER VECTOR FORMATION

GMM-UBM is developed for deriving a target speaker’s GMM by adapting the parameters of the model. Only the mean vectors of target speaker
model are adapted using Equations (8.5) to (8.9). Specifically, given an enrolment utterance with acoustic vector sequence \( X = \{ \tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_r \} \), the mean vectors \( \mu_i \) of the UBM is used to obtain the adapted mean vectors

\[
\hat{\mu}_i = \alpha_i E_i(X) + (1-\alpha_i) \mu_i, \quad i = 1, ..., M,
\]

where

\[
\alpha_i = \frac{n_i(X)}{n_i(X) + r}, \quad (8.5)
\]

\[
n_i(X) = \sum_{t=1}^{r} \Pr \left( \tilde{x}_t \right), \quad (8.6)
\]

\[
E_i(X) = \frac{1}{n_i} \sum_{t=1}^{r} \Pr \left( \tilde{x}_t \right), \quad (8.7)
\]

\[
\Pr \left( \tilde{x}_t \right) = \frac{\lambda_i p_i}{\sum_{j=1}^{M} \lambda_j p_j} \left( \hat{\mu}_i \right), \quad (8.8)
\]

The adapted means vectors of all \( M \) mixtures are combined to produce the GMM super vector

\[
\hat{m}_{[1 \times MD]} = [\hat{\mu}_1, \hat{\mu}_2, ..., \hat{\mu}_M]; \quad (8.10)
\]

The GMM super vector can be thought of as a mapping between an utterance and a high dimensional vector.

As stated in (Man Wai Mak & Wei Rao 2011), the number of super vectors for target speakers and impostors is increased by means of
partitioning the utterance into groups. Then, for each group a super vector is created and this is repeated several times by randomly rearranging the sequence of utterance. This method of utterance partitioning provides sufficient number of super vectors for training the SVM.

8.5 FUZZY SUPPORT VECTOR MACHINE

In standard SVM, training vectors are labelled with either +1 or -1 to indicate the class in which it belongs. Training vectors of all classes are grouped into two classes with equal weight. There are two different ways in grouping training vectors (Fuqian Shi & Jiang Xu 2012) (1) One versus One (2) One versus rest. During training of SVM for a target speaker, in Equation (9.2), the $\alpha$, $b$ and support vectors are optimized to produce a model. Among the optimum support vectors, some training vectors may have more importance than others. Fuzzy theory deals with these issues by saying that a training vector $x_i$ belongs 80% to class +1 and 20% to class -1. This may be achieved by associating a fuzzy membership value $0 \leq \mu_i \leq 1$ with each training pair $(x_i, t_i)$ (Lin & Wang 2002). This membership value represents that the vector belongs to the class $t_i$ with membership value $\mu_i$ and to class $t_{ja}$ with membership value $1 - \mu_i$. Support vectors those have less contribution in the learning process are neglected based on the membership value. Thus the number of support vectors used in the representation of speaker model is reduced and its performance will be equivalent to classical SVM even with noisy data points.

FSVM based speaker verification system is proposed. Let X be an input space and T be an output space. Each training pattern is given a label $t_i$ from T and a fuzzy membership value $\sigma \leq \mu_i \leq 1$ with $i = 1,...,l$. Value of $\sigma$ must be sufficiently small but greater
than zero (Chun-fu Lin & Sheng-de Wang 2004) i.e., Since the fuzzy membership value \( \mu_i \) gives information about the percentage of corresponding data point \( x_i \) in the class \( t_i \), FSVM requires a parameter \( \xi_i \) to measure the error in the SVM. The product \( \mu_i \xi_i \) is the measure of error with different weighting. The primal hyper plane problem in FSVM is stated as:

\[
\min_{w,b,\xi \in \mathbb{R}^n} L(w,b,\xi) = \frac{1}{2} w^T w + C \mu^T \xi
\]

\[
t_i (w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, I,
\]

such that: \( \xi_i \geq 0 \).

where \( C \) is a regularization parameter which is used to balance between the minimization of the error function and the maximization of the margin of the optimal hyper plane;

\( w \) is the weight vector;

\( b \) is the bias;

\( \xi_i \) is the slack variable for the data point which is not fitted in the optimal hyper plane.

Since the training pair \((x_i, t_i)\) is weighted by the membership value \( \mu_i \), the training pair with less \( \mu_i \) will have less influence in the decision surface than those with larger \( \mu_i \) value.

Solving Equation (8.11) is a Quadratic Programming problem (Burges & Schölkopf 1997). This can be solved by transforming the problem into dual problem as
\[
\max W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j t_{ij} K(x_i, x_j)
\]

Subject to \[\sum_i t_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \mu_i C, \quad i = 1, \ldots, l\]

where \(\alpha_i\)'s are non negative Lagrangian multipliers;

\[K(x_i, x_j)\] is Kernel function which provides the distance between two data points in feature space, i.e.,

\[K(x_i, x_j) = \phi(x_i), \phi(x_j)\]

where \(\phi(x_i)\) is the mapping function from input space to feature space. Then the optimal hyper plane is defined by decision function

\[g(t) = \sum_i \alpha_i K(x_i, x_j) + b\]

where \(b = t_i - \sum_i \alpha_i K(x_i, x_j)\) \(\forall j: 0 < \alpha_j < C\)

**Figure 8.2 Automatic Speaker Verification system with Fuzzy SVM**
and the Kuhn Tucker conditions are defined as

$$\alpha_i (t_i (wz_i + b) - 1 + \xi_i) = 0, \quad i = 1, \ldots, l. \quad (8.17)$$

$$\left( \mu_i C - \overline{\alpha_i} \right) \tilde{\xi}_i = 0, \quad i = 1, \ldots, l. \quad (8.18)$$

During training, the membership values are modified such that it can control the tradeoffs between the maximization of margin and the amount of misclassification.

### 8.6 Gaussian Mixture Model with Fuzzy Support Vector Machine

In GMM-FSVM as shown in Figure 8.2, when the super vector of test utterance is given as the input to FSVM of target speaker, the verification score of the claimed speaker is given by

$$score^c = \alpha^c K(x^c, x^t) - \sum_{i \in S^b} \alpha_i^b K(x^c, x^b) \quad (8.19)$$

where $\alpha^c, \alpha^b$ are Lagrangian multipliers of claimed speaker and background speakers respectively;

$x^c, x^t, x^b$ are claimed speaker’s super vector(test super vector), target speaker’s super vector and background speakers’ super vectors respectively;

$K \cdot, \cdot$ is a kind of distance measure between two super vectors in the high dimensional feature space.
Experiments have been performed for GMM-SVM based system with various kernel functions such as, Linear, Polynomial, Radial, Quadratic and RBF.

Polynomial kernel is given by

\[ K(x^c, x') = (x^c^T x' + 1)^n \] (8.20)

where \( n \) is the polynomial order.

Radial Kernel is given by

\[ K(x^c, x') = e^{\frac{1}{\sigma^2} \left\| x^c - x' \right\|^2} \] (8.21)

where \( \sigma \) is the width of the radial basis function.

Quadratic kernel is given by

\[ K(x^c, x') = (x^c^T x' + 1)^2 \] (8.22)

An attractive feature of the SVM (Campbell 2004) is that the selection of sub clusters is implicit, with each support vectors contributing one local Gaussian functions, and centered at that data point. By the kernel function, the super vectors are positioned on the surface of hyper sphere in the feature space. Then \( K(x_i, x_j) = \phi(x_i), \phi(x_j) \) is the cosine of the angle between \( \phi(x_i) \) and \( \phi(x_j) \). An FSVM is trained for each target speaker using the GMM super vector of the speaker’s enrolment utterances as positive samples, and GMM super vectors of all utterances from background speakers as negative samples.
8.7 EXPERIMENTAL SETUP

8.7.1 Database

Speaker verification experiments were conducted using the NIST 2001 SRE database. The NIST 2001 SRE development database consists of 38 male speakers and 22 female speakers. The evaluation database comprises 74 male speakers and 100 female speakers for training, 850 male speakers and 1188 female speakers for testing. The training utterance for each speaker was for 2 minutes and the testing segment duration was less than 60 seconds. Development database is used for model development and initial validation whereas the evaluation database is used for final validation. Equal Error Rate (EER) and the minimum Decision Cost Function (minDCF) are used as metrics for performance evaluation.

8.7.2 Feature Extraction and Feature Warping

In this work, we extracted 13-dimensional Mel frequency Cepstral Coefficients from speech signal for 30ms duration with 20ms overlapping. Cepstral Mean Subtraction (CMS) and RelAtive SpecTrAl (RASTA) filtering are two of the standard feature-based channel compensation techniques. But even after CMS and RASTA filtering, channel and handset mismatch can still cause lots of errors. Hence, with CMS and RASTA, recently introduced feature warping technique called Gaussianization is also used to transform the distribution of a cepstral coefficient feature stream to a Normal distribution based on Cumulative Distribution Function (CDF). It is shown that this technique has brought about significant improvements in recognition rate of the system compared to system based on standard techniques. Then first order and second order deltas are appended to the Gaussianized cepstral vector. The size of the feature vector is now 39.
8.7.3 GMM-SVM and GMM-FSVM Model

In GMM-SVM & GMM-Fuzzy SVM (FSVM) based systems, the super vectors are formed from the mean vectors of MAP adapted GMMs. In super vector formation, if an entire utterance is used to develop a speaker specific model through MAP adaptation, it yields only one super vector per speaker. But it is not enough for training and testing the SVM based speaker model. Hence in this work, Utterance partitioning method proposed by Man Wai Mak & Wei Rao (2011) is followed. Training utterances of target and background speakers are partitioned into five groups and a super vector is formed for each group. That means five super vectors per speaker (includes target speaker (17) and background speakers (80) in the case of development) are obtained by dividing the training utterance into five sub utterances. These super vectors (97x5=487) are used as negative training vectors for FSVM. The positive training vectors (for target speakers) are obtained by dividing the training utterance into 10 sub utterances. For each sub utterance a super vector is formed from the GMM trained with these sub utterances. This is repeated 20 times by randomly rearranging the sequence of utterance and every time dividing it into 10 groups. Thus for every target speaker there will be 200 super vectors plus one super vector for full utterance. FSVM is trained for each target speaker using 201 super vectors as positive class data and 487 super vectors as negative class data. The same procedure is followed for SVM based system also. Impostor’s super vectors are formed by dividing the training utterance of each imposter into 10 sub utterances and one super vector from full utterance. Similarly for testing the target speaker, the test utterance is subdivided into 10 groups and a super vector for each group plus one full utterance super vector are obtained. So, totally, 11 super vectors are used for testing each target speaker.
The super vectors are normalized for compensating intersession variability using Fisher’s Linear Discriminent Analysis (FLDA). In FLDA, the super vectors are transformed from high dimensional feature space into low dimensional feature space; also it provides the principal directions of channel variability. The Eigen space is generated using both target and background speakers super vectors. Then target, background, impostor and test super vectors are projected into this Eigen space. The projected super vectors are used for training and testing the SVM and also FSVM.

8.8 SUMMARY

In this chapter, the SVM based speaker verification system is analysed with various kernel functions. The super vectors are computed from the GMM and they are used as the input for the SVM system. In order to improve the performance of the SVM based system, the fuzzy based SVM system is also developed and analysed.