

Feature Extraction and Dimensionality Reduction in SVM Speaker Recognition

Thembisile Mazibuko, thembi@crg.ee.uct.ac.za Daniel J. Mashao, daniel@ebe.uct.ac.za

Department of Electrical Engineering, University of Cape Town

Abstract—The Support Vector Machine is a discriminative classifier which has achieved impressive results in several pattern recognition tasks. Its applicability is however limited by the computational expense. The SVM training and testing times increases with increasing amounts of data. A possible solution is the reduction of the number of computations within the SVM kernel by reducing the dimensionality of the observation vectors. In this paper we apply this concept of reducing the data dimensionality to decrease the complexity to an SVM-based speaker verification task, carried out on a subset of the NIST 2000 speaker recognition evaluation. The dimensionality reduction is performed by applying Principal Component Analysis to the feature vector. The results show degradation in the performance of SVM when feature extraction is applied. There is however also a significant decrease in the training and testing times.

Key Terms—speaker verification, support vector machine, feature transformation

AUTOMATIC speaker recognition is the task of using machines to recognize people using speech as a biometric.

The speech input is converted into a feature vector representation. Linear Predictive Coefficients (LPC) and Mel-Frequency Cepstral Coefficients (MFCC) are popular feature sets. These feature sets are used to create models which represent each speaker. In the testing phase the input speech is compared to the model and a classification engine is used to decide who the test speaker is.

The Support Vector Machine (SVM) is a discriminative classifier which has been successfully applied different pattern recognition fields. In the speech processing field the authors of [1] applied SVM to a speaker and language recognition task on the NIST 2003 database with good results. In [2] the author applied SVM to a speaker verification task and found that when combined with a Gaussian Mixture Model (GMM) classifier, SVM improved on the performance of the more popular classifier. The power of SVM lies in the implicit transformation of the input space to a higher dimensional space. Because the transformation is performed implicitly, there are no computations explicitly performed in the high dimensional space. Intuitively, this means that the dimension of the input dataset does not affect the performance of SVM. However, according to [3], the performance of SVM is dependent on the radius of the data, which increases with the number of features. A major hindrance in the application of SVM is that the classifier suffers from extremely long training time especially as the size of the data, or number of observation vectors, increases.

Research studies have shown an improvement in the accuracy of discriminative classifiers by applying feature extraction techniques [4,5]. The basis of these studies is that by applying

feature extraction we can remove redundancies in the feature vectors by extracting and using only those features which are most relevant.

Principal component analysis (PCA) is a relatively old and well developed linear independent feature extraction technique which has been applied to several pattern recognition tasks. Pattern recognition tasks are divided into two phases; feature analysis and classification. Feature extraction is a part of the feature analysis phase where we attempt to reduce redundancy in the feature vectors. An attraction of techniques likes PCA (others include Linear Discriminant Analysis and Independent Component Analysis) is their data dimensionality reduction capabilities.

In [5] the author proposed what he called the Reduced Dimensional Support Vector Machine which applies feature extraction and dimensionality reduction techniques. The concept behind RDSVM is that the computational burden on SVM classification can be decreased by reducing the number of computations within the classifier. It is generally not possible nor desirable to reduce the number of observation vectors as this might lead to loss of important information. However, SVM performance suffers from long training and testing times when the number of observation vectors is too high. So, RDSVM reduces the dimensionality of the observation vectors as a way of reducing the total number of SVM computations.

In the experiments discussed in this paper, the effect of feature extraction on the system accuracy and speed of SVM training and classification on a speaker verification task was studied. The rest of this paper is organized as follows, Section I gives an introduction to the SVM classifier. In Section II we present the mathematical formulation of Principal Component Analysis. In Section III a discussion of speaker recognition is given. Section IV is the experimental setup and the results of the experiments conducted. Section V is a discussion of the results and Section VI is the conclusion and a short discussion of future work.

I. SUPPORT VECTOR MACHINE CLASSIFIER

The Support Vector Machine is a powerful discriminative classifier which maps the input onto a high dimensional space by $\Phi : R^d \Rightarrow \mathcal{S}$; and then finds an optimal hyperplane to separate the data in that space. This separating hyperplane is found by maximizing the distance of the closest patterns [6]. The new space is often referred to as the feature space. Here we will present a simplified, general explanation of SVM.

Suppose we have a binary classification problem as shown in figure 1 where each example belongs to either class +1 or -1. SVM seeks to maximize the margin between the two classes by finding the separating hyperplane which lies halfway between the data classes.

We can, without loss of generality, consider the case of data that are *non-linearly* separable. When the data are transformed by some non-linear transformation onto a higher dimensional space, they ‘spread out’ allowing a separating hyperplane to be found in the feature space.

Figure 1: Finding the separating hyperplane for non-linearly separable data

The *support vectors* are those data points that lie on S_1 and S_2 , lie within the margin, or are misclassified. The hyperplane in the high dimensional transform space results in a complex decision surface in the input data space. SVM allows for these misclassifications in training with a user-defined cost parameter C . This way the misclassifications are limited while still avoiding over-fitting the training data [2,7].

In general the SVM is computed by using the *kernel trick* so

$$f(x) = \sum_{i=1}^N \alpha_i y_i K(s_i, x) = b \quad (1)$$

where K is some kernel function such that

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2)$$

and s_i are the support vectors.

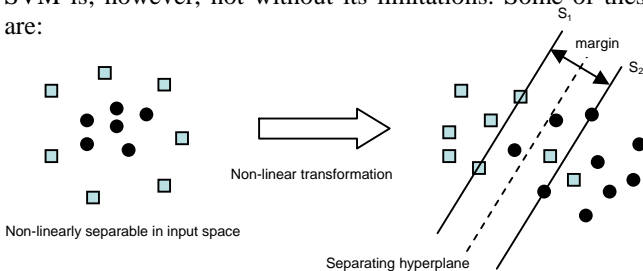
a.) SVM Characteristics

A distinguishing characteristic of SVM is its strong foundation on statistical learning theory which establishes a bound on the generalization error (the error rate of a learning machine on unseen data) thus improving the classification results for unseen patterns [6]. The SVM minimizes this bound by maximizing the margin.

Also, since SVM projects the input onto a higher dimension space, the margin maximization is independent of the original dimension of the input space, thus SVM successfully avoids the curse of dimensionality which some classifiers suffer from.

b.) Challenges in SVM

SVM is; however, not without its limitations. Some of these are:



- ◆ Determining most appropriate choice of kernel for a particular task.
- ◆ Once a kernel has been chosen, there still remains the issue of optimizing the parameters of the kernel. For instance, in [8] the authors conclude that *there is an optimum C value for each dataset*.
- ◆ The design of SVM is optimized for binary classification which can limit its applicability to multi-class classification tasks.
- ◆ Although the SVM successfully avoids the *curse of dimensionality*, the technique can be incredibly slow, especially as the size of the data increases. This is true for both the training and test phases.

There are still several challenges which need to be addressed in SVM applications. In [7] the author suggested that there might be a possibility for limiting the computational load of SVM by reducing the data dimensionality which would decrease the number of computations that have to be performed within the SVM. The following section presents an overview of a popular dimensionality reduction technique.

II. PRINCIPAL COMPONENT ANALYSIS

A possible approach to improve the classification performance of SVM is to operate the classifier in a feature space in which the classes are inherently separated [7]. This feature space is typically a multi-dimensional space resulting from transforming the input space via some linear or non-linear transformation. Principal Component Analysis (PCA) is one such transform.

PCA is also referred to as the discrete Karhunen-Loève Transform (KLT) or the Hotelling transform. The central principle in PCA is to transform the input space onto a feature space where the data show maximal variance.

The following is a brief discussion of the mathematical formulation of PCA. The PCA formulation given here closely follows that of [2,5].

Let \mathbf{X} be an m -dimensional input data vector such that $x = (x_1, \dots, x_m)^T$. We estimate the sample mean by

$$\mu \approx \frac{1}{N} \sum_{i=1}^N (x_i) \quad (3)$$

where N is the number of samples. The sample covariance matrix \mathbf{C} becomes

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^T (x_i - \mu) \quad (4)$$

$$= \frac{1}{N} \sum_{i=1}^N x_i x_i^T - \mu \mu^T \quad (5)$$

To perform PCA we find the eigenvalues and eigenvectors of the sample covariance matrix. Rearranging the eigenvectors in descending order according to the corresponding eigenvalues, a linear transformation matrix \mathbf{T} is formed which generates new vectors from \mathbf{X} by

$$x' = T(x - \mu) \quad (6)$$

The eigenvectors of \mathbf{C} are the *principal components*. In the projected space, the new vectors \mathbf{X}' are minimally correlated.

In order to exploit the dimensionality reduction of PCA we would simply choose the top k ($k < m$) eigenvectors of \mathbf{C} to form \mathbf{T} . This is the common way of choosing the eigenvectors to include in the transformation matrix. An assumption that is made in PCA dimensionality reduction is that most of the information contained in the observation vectors can be adequately represented in the subspace spanned by the first k principal components.

a.) Shortcomings of PCA

PCA is a well-established technique in pattern recognition and research into its application is worthwhile.

III. AUTOMATIC SPEAKER RECOGNITION

Automatic speaker recognition is task of using a computer to determine who an utterance was spoken by using a sample of their speech as a biometric measure. Speaker recognition divides into two categories, speaker identification and speaker verification. In this paper we are concerned with the latter which, as the name suggests, is the process of authenticating whether a speaker is who they claim to be. The degree of similarity between the test speech sample and the claimed speaker's model is compared to some predefined threshold and the result used to decide whether to accept or reject the claimant. This task then also divides into text-dependent and text-independent speaker verification.

In text-independent speaker verification the content of the speech used for the verification is *a priori* unknown. This type of authentication system is more suited to highly security sensitive applications as the risk of an impostor finding out the required text and gaining access to the system.

Figure 2 shows a simplified speaker verification system. The detailed discussion of all the components of the system is beyond the scope of this paper. However, the interested reader is referred to [9,10].

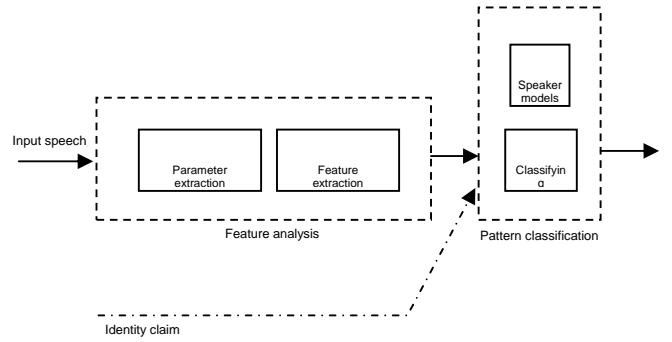


Figure 2: A simplified speaker verification system

The speaker verification is a two class problem in that we are trying to determine whether the test speech belongs to the claimed speaker (class +1) or to an impostor set (class -1). The impostor set is generally approximated by speech data from several possible impostors. A test utterance is compared to the speakers' model and the impostor set and the classifier then decides whether the claimant is who they claim to be or an impostor.

a.) Measuring System Performance

In speaker verification there are two types of errors that may occur; *false acceptance* (FA) and *false rejection* (FR).

An FA error refers to the case when an impostor is classified as a authentic system user. Applications that require high security aim to keep these errors at a minimum in order to protect the system from unauthorized use. The FR error occurs when an authorized user is wrongly classified as an impostor and thus denied access to the system. There is generally a tradeoff between FA and FR errors. There a common way of measuring the performance of a speaker verification system is by defining the point where rate of FR's is equal to the rate of FA's. This point is known as the Equal Error Rate (ERR).

The Detection Error Tradeoff (DET) [11] curve is a popular way of graphically representing the expected performance of speaker verification system. The curve is a plot of the rate of the errors mentioned above, on a deviate scale. An advantage of the DET curve is that it lends itself to easy interpretation, the close the system's DET curve moves to the origin, the better the system performance.

IV. EXPERIMENTAL SETUP AND RESULTS

As mentioned, speaker verification is a two class problem. This suggests that SVM, which is inherently a two-class classifier, would be a natural choice for this task. It has already been successfully applied to this task in [1,2,12]. In this section present the experimental setup and a discussion of the results obtained.

a.) Experimental Setup

For the purposes of our study we followed very closely the experimental setup used in [1] when building up our baseline.

That is, we used 12 linear predictive coefficients (LPC) from which 18 cepstral coefficients (LPCC) and their deltas were computed which resulted in a 36-dimensional feature vector. A simple, energy based voice activity detection was used to remove those frames with energy levels below a certain threshold and mean and variance normalization was applied. We did however make use of a RBF kernel function for the SVM classifier which differs from [1]. The SVM classifier used is part of the Torch machine learning library from the IDIAP Research Institute.

The speaker verification experiments were carried out on the NIST 2000 database. The impostor model was created using the NIST 1999 database so that none of the testing data was used in the impostor model. This way we avoid introducing any sort of bias to the system. All the experiments were conducted on a 3.2GHz Pentium 4 processor.

b.) Results

The results below compare the performance of the baseline system (which uses a 36 dimensional feature vector) to the results achieved when Principal Component Analysis feature extraction and dimensionality reduction was applied.

TABLE 1: Results of feature extraction and dimensionality reduction on SVM

Speaker Verification System	Average Training Time [seconds]	Average Testing Time [seconds]
Baseline	11538.45	12.12
36-d PCA	7786.45	9.92
32-d PCA	244.39	4.22
24-d PCA	300.57	4.90

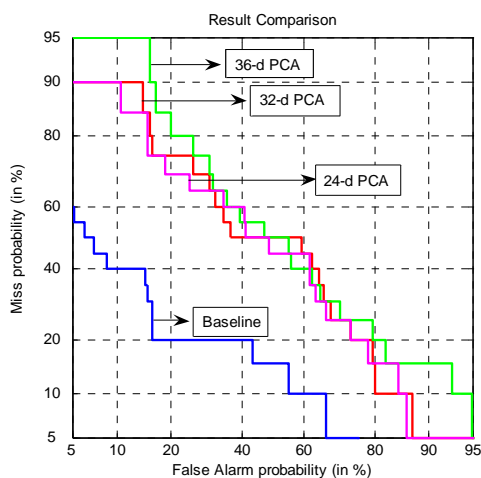


Figure 3: DET plot comparing results

V. DISCUSSION OF RESULTS

Table 1 shows the average training and testing times in seconds. The baseline system had the longest training time while the systems which employed PCA feature extraction showed a significant decrease in both training and testing times.

Figure 3 shows the resultant DET curves. The baseline outperforms the application of PCA. A possible reason of this could be that by applying the PCA transformation we change the structure of the data. As mentioned, the choice of kernel and kernel parameters for SVM is data dependent. Thus it is likely that the transformed data could be better classified by a different kernel function.

VI. CONCLUSIONS AND FUTURE WORK

The results show that applying feature extraction, even without dimensionality reduction, decreases the training and testing times. However there is also degradation in performance when PCA is applied. The savings in processing time make it worthwhile to explore the possibility of improving the system performance while applying feature extraction. A few possibilities for achieving are noted below.

The choice of optimal kernel for any particular SVM task remains a matter of trial and error. There is no formula to determine which kernel is most appropriate for which task. Thus it may be necessary to change the kernel parameters or maybe even the kernel itself in order to obtain optimum performance on data that has been transformed by PCA or any other transformation algorithm.

In this study we chose to use the traditional approach of using the eigenvalues as the criteria for choosing the eigenvectors with which to build the transformation matrix for PCA. In future work we aim to repeat the experiment, having optimized the system parameters, using the Fisher ratio in order to determine whether this method will perform as well on an SVM based system as it did on the VQ based system in [4].

Other future work includes investigating the performance of other feature extraction techniques such as Independent Component Analysis when applied to this task.

VII. REFERENCES

- [1] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer and P. A. Torres-Carrasquillo, "Support Vector Machines for Speaker and Language Recognition". *Computer Speech and Language*, August 2005.
- [2] V. Wan, "Speaker Verification with Support Vector Machines". PhD thesis, Department of Computer Science, Department of Sheffield, June 2003.
- [3] L. Wolf and S. Bileschi, "Combining Variable Selection

with Dimensionality Reduction”. Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, CBCL Memo 247, March 2005

[4.] P. Ding and L. Zhang, “Speaker Recognition using Principal Component Analysis”. In *Proceedings of ICONIP 2001*. Shanghai, China. November 2001

[5.] X. Wang, “Feature Extraction and Dimensionality Reduction in Pattern Recognition and Their Application in Speech Recognition”. PhD thesis, School of Microelectronic Engineering, Griffith University, November 2002.

[6.] M. Awad and L.Khan, “Applications and Limitations of Support Vector Machines”. Department of Computer Science, University of Texas at Dallas, USA.

[7.] A. Ganapathiraju, “Support Vector Machines for Speech Recognition.” PhD thesis, Department of Electrical and Computer Engineering, Mississippi State University, May 2002.

[8.] P. Watanachaturaporn, P.K. Varshney and M. K. Arora, “Evaluation of Factors Affecting Support Vector Machines for Hyperspectral Classification”. In *Proceedings of ASPRS 2005*. Baltimore, USA, March 2005.

[9.] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”. *Data Mining and Knowledge Discovery*, 1998, vol.2, pg 121-167.

[10.]F. Bimbot, J-F. Bonastre, C. Fredouille, G. Gravier, I Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacrétaz and D. A. Reynolds, “A Tutorial on Text-Independent Speaker Verification”. *Journal on Applied Signal Processing* 2004 vol 4, pg 430-451.

[11.]A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, “The DET Curve in Assessment of Detection Task Performance”.

[12.]W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones and T. R. Leek, “Phonetic Speaker Recognition with Support Vector Machines”. MIT Lincoln Laboratory, Lexington.

[13.]M.E. Wall, A. Rechtsteneir and L.M. Rocha, “Singular Value Decomposition and Principal Component Analysis.” In *A Practical Approach to Microarray Data Analysis*, (editors D.P. Berrar, W. Dubitzky and M. Granzow. Kluwer; Norwell, MA, 2003, pg 91-109.

[14.]R. Collobert and S. Bengio, “SVM Torch: Support Vector Machines for Large-Scale Regression Problems”. *Journal of Machine Learning Research*, 2001, vol. 1, pg 143-160.
