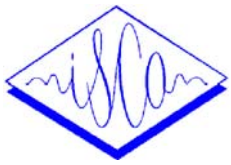

ODYSSEY04 - The Speaker and Language Recognition Workshop
Toledo, Spain
May 31 - June 3, 2004

Language Recognition with Support Vector Machines*

W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, D. A. Reynolds

MIT Lincoln Laboratory
Lexington, MA USA
{wcampbell,es,ptorres,dar}@ll.mit.edu

Abstract

Support vector machines (SVMs) have become a popular tool for discriminative classification. Powerful theoretical and computational tools for support vector machines have enabled significant improvements in pattern classification in several areas. An exciting area of recent application of support vector machines is in speech processing. A key aspect of applying SVMs to speech is to provide a SVM kernel which compares sequences of feature vectors–a sequence kernel. We propose the use of sequence kernels for language recognition. We apply our methods to the NIST 2003 language evaluation task. Results demonstrate the potential of the new SVM methods.

1. Introduction

Many successful approaches to language recognition have been proposed. A classic approach implemented in the parallel-phone recognition language modeling (PPRLM) system of Zissman [1] used phone tokenization of speech combined with a phonotactic analysis of the output to classify the language. A more recent development is the use of methodologies similar to those in speaker recognition. In these approaches, a set of features useful for language recognition have been combined with the ubiquitous Gaussian mixture model to produce excellent recognition performance [2, 3].

We adopt the methods of [2] using a purely acoustic approach (i.e., no intermediate representations such as phone labels). For input features, we use shifted delta cepstral coefficients. For classification, we use unique support vector machine (SVM) methods designed for operating on sequence data [4].

The organization of our paper is as follows. In Section 2, we discuss the training and testing scenario for the recent NIST Language Recognition evaluation (2003). Section 3 describes our feature set. In section 4, we briefly review SVMs and discuss our approach to language recognition. Section 4.3 discusses the sequence kernel used in more detail. Section 5 presents the fusion approach applied. Finally, Section 6 discusses experiments and results using the new system.

2. 2003 NIST Language Recognition Evaluation

In 2003, NIST held an evaluation to assess the current performance of language recognition systems for conversational telephone speech. The basic task of the evaluation was to detect the presence of a hypothesized target language given a segment of speech. The target languages were American English, Arabic, Farsi, Canadian French, Mandarin, German, Hindi, Japanese, Spanish, Korean, Tamil, and Vietnamese. Evaluation of the task was performed through standard measures–a decision cost function and equal error rate.

The training, development, and test data was primarily drawn from the CallFriend corpus available from the Linguistic Data Consortium (LDC). Training data consisted of 20 complete conversations (nominally 30 minutes) for each of the 12 target languages. Development data was drawn from the 1996 NIST LID development and evaluation sets. Test data consisted of speech segments of length 3, 10, and 30 seconds. For each of these durations, 1,280 utterances were available; this resulted in 15,360 detection trials per duration. For more information, we refer to the NIST evaluation plan [5, 6].

3. Features for Language Recognition

One of the breakthroughs for performing language recognition using Gaussian mixture models was the discovery of a better feature set for language identification [2]. The improved feature set, shifted delta cepstral (SDC) coefficients, are an extension of delta-cepstral coefficients. Prior to the use of SDC coefficients, GMM-based language recognition was less accurate than alternate approaches [1].

SDC coefficients are calculated as shown in Figure 1. SDC coefficients are based upon four parameters, typically written as N-d-P-k. For each frame of data, MFCCs are calculated based on N; i.e., $c_0, c_1, \ldots, c_{N-1}$ (note that c_0 is used). The parameter d determines the spread over which deltas are calculated, and the parameter P determines the gaps between successive delta computations. I.e., for a given time, t, we obtain

$$\Delta c(t,i) = c(t + iP + d) - c(t + iP - d) \qquad (1)$$

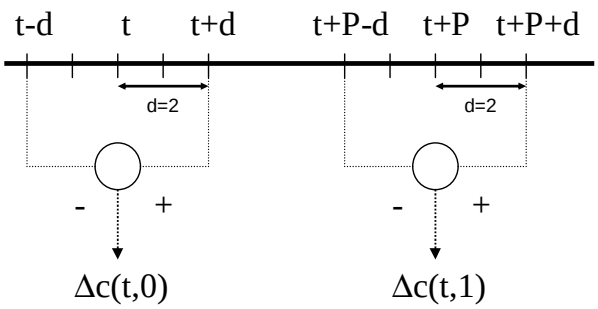

Figure 1: Shifted delta cepstral coefficients

*This work was sponsored by the Department of Defense under Air Force contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

as an intermediate calculation. The SDC coefficients are then a stacked version of (1),

$$SDC(t) = \left[\Delta c(t,0)^t \quad \Delta c(t,1)^t \quad \ldots \quad \Delta c(t,k-1)^t\right]^t. \tag{2}$$

4. Support Vector Machines for Language Recognition

4.1. Support Vector Machines

A support vector machine (SVM) [7] is a two-class classifier constructed from sums of a kernel function $K(\cdot,\cdot)$,

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + b; \tag{3}$$

where the t_i are the target values, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization process [8]. The target values are either 1 or -1 depending upon whether the corresponding support vector is in class 0 or class 1. For classification, a class decision is based upon whether the value, $f(\mathbf{x})$, is above or below a threshold.

The kernel $K(\cdot,\cdot)$ is constrained to have certain properties (the Mercer condition), so that $K(\cdot,\cdot)$ can be expressed as

$$K(\mathbf{x},\mathbf{y}) = \mathbf{b}(\mathbf{x})^t \mathbf{b}(\mathbf{y}) \tag{4}$$

where $\mathbf{b}(\mathbf{x})$ is a mapping from the input space (where $\mathbf{x}$ lives) to a possibly infinite dimensional space.

The optimization condition relies upon a maximum margin concept, see Figure 2. For a separable data set, the system places a hyperplane in a high dimensional space so that the hyperplane has maximum margin. The data points from the training set lying on the boundaries (as indicated by solid line in the figure) are the support vectors in equation (3). The focus then of the SVM training process is to model the boundary as opposed to a traditional Gaussian mixture model which would model the probability distributions of the speaker (a generative approach).

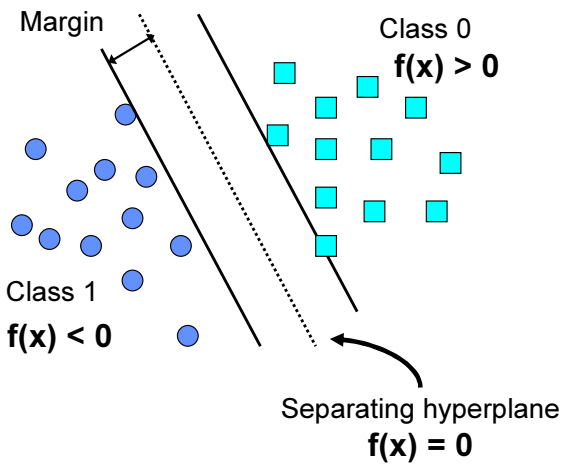

Figure 2: Support vector machine concept

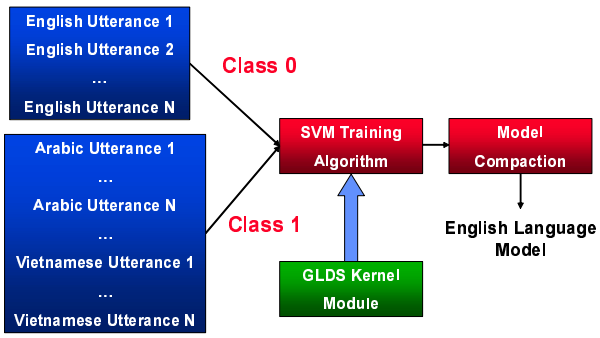

Figure 3: Training strategy

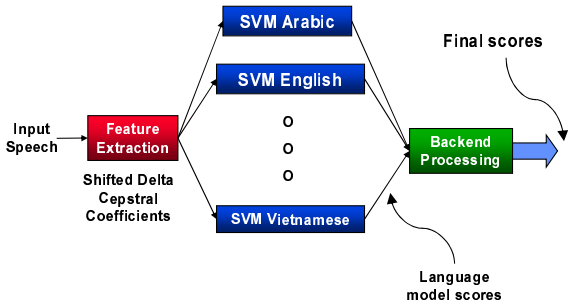

Figure 4: Language recognition using SVMs

4.2. SVMs for Language Recognition

In order to use support vector machines for language recognition, two issues must be resolved. First, we must have a framework for using the SVM as a multiclass classifier. Second, the SVM must be able to handle sequences of feature vectors (i.e., sequences of SDC coefficients) in order to perform classification.

The first issue, handling multiclass data, is straightforward. We use a "one vs. all" strategy as shown in Figure 3. The figure shows an example of training for an English model. In the figure, we use English for class 0 data, and the remaining languages are used for class 1 data. This training data is processed with a standard SVM optimizer (we have used SVMTorch [8]) using our kernel, the GLDS (Generalized Linear Discriminant Sequence) kernel. The resulting SVM model is reduced in size and that process produces an English language model. For other languages, the process is analogous.

After obtaining the target language models, we can then combine them together to produce a language recognition system. The result is shown in Figure 4. Speech is input into the SVM language recognition system, shifted delta cepstral features are extracted. Then an SVM for each of the 12 language models is applied to the feature vector sequence each producing a score (a single number). These scores are then processed using a backend fusion system [2].

4.3. Sequence Kernels

A second issue for successful SVM language recognition implementation is the ability of the system to handle sequence data. Our solution is to implement *sequence kernels*. That is, we construct a kernel, $K(\{\mathbf{x}_i\},\{\mathbf{y}_i\})$ that compares two sequences of

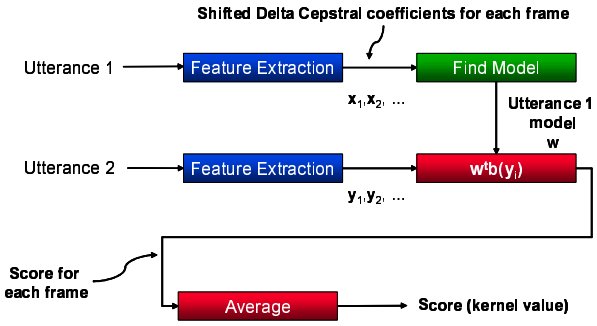

Figure 5: Sequence kernel

feature vectors, $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$. The main issues in constructing a kernel are (1) to satisfy the Mercer condition, and (2) make the kernel a relevant comparison between two sequences.

Our basic approach to kernel construction is shown in Figure 5. To compare two utterances, we first perform feature extraction to produce two sequences of feature vectors. From the first utterance, we find a model using *only* the data from the utterance. One might think this is an ill-conditioned process, but typically this is done by adapting from a "background model." After finding a model $\mathbf{w}$ from the first utterance, this model is then used to score the second utterance using a generalized linear discriminant, $\mathbf{w}^t\mathbf{b}(\mathbf{y}_i)$. The output is a score which is our kernel value.

The generalized linear discriminant $\mathbf{w}^t\mathbf{b}(\mathbf{y}_i)$ is simply an inner product between a model and a set of basis functions, $b_j(\cdot)$; i.e.,

$$\mathbf{b}(\mathbf{y}_i) = \begin{bmatrix} b_1(\mathbf{y}_i) & \cdots & b_M(\mathbf{y}_i) \end{bmatrix}^t. \tag{5}$$

We choose monomials up to a given degree for the basis functions in this paper (an example basis function would be $y_{i,1}^2 y_{i,2}^3 y_{i,5}$). Other bases such as radial basis functions, etc., could be used.

The kernel resulting from Figure 5 can be written as

$$K_{\text{GLDS}}(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) = \bar{\mathbf{b}}_x^t \bar{\mathbf{R}}^{-1} \bar{\mathbf{b}}_y. \tag{6}$$

The mapping $\{\mathbf{x}_i\} \rightarrow \bar{\mathbf{b}}_x$ is defined as

$$\{\mathbf{x}_i\} \rightarrow \frac{1}{N_x} \sum_{i=1}^{N_x} \mathbf{b}(\mathbf{x}_i), \tag{7}$$

and $\bar{\mathbf{b}}_y$ is defined in an analogous manner to (7). $\bar{\mathbf{R}}$ is a correlation matrix derived from a large background population. We refer to (6) as the Generalized Linear Discriminant Sequence (GLDS) kernel [4]. Note that (6) satisfies the Mercer condition since we have an inner product.

5. Fusion

Fusion was performed using a backend classification system commonly used in language recognition systems [1, 9]. The backend system consists of three parts (applied in the listed order): a feature transformation component, a set of Gaussian classifiers (equal to the number of target languages), and finally a log-likelihood ratio (LLR) normalization. The backend takes the target language scores from all available classifiers and maps them to 12 target language scores.

The feature transformation is trained from the development set included in the NIST distribution. Linear discriminant analysis (LDA) has been used for this task. The Gaussian classifiers are also trained from the development data; a global covariance is used.

As a last step in the backend, the scores are converted to log-likelihood ratios. Suppose $s_1, \ldots, s_M$ are the scores from the M language models for a particular message. To normalize the scores, we find new scores, s_i' given by

$$s_i' = s_i - \log_{10}\left(\frac{1}{M-1}\sum_{j\neq i} 10^{-s_j}\right) \tag{8}$$

6. Experiments

Experiments were performed using the NIST LRE evaluation data and the primary evaluation condition. We focus on language detection for the 30 second case. This resulted in 960 true trials and 10,560 false trials.

For the SVM system, SDC features were extracted as in Section 3. Our primary representation N-d-P-k was 7-1-3-7. This representation was selected based upon prior excellent results with this choice [2, 9]. After extracting the SDC features, non-speech frames were eliminated, and each feature was normalized to mean 0 and variance 1 on a per utterance basis. This resulted in a sequence of features vectors of dimension 49 for each utterance.

The SVM system used the GLDS kernel as described in Section 4.3 with a diagonal covariance matrix $\bar{\mathbf{R}}$. All monomials up to degree 3 were used in the expansion $\mathbf{b}(\mathbf{x})$; this resulted in an expansion dimension of 22,100.

For contrast, we compare our SVM system to a Gaussian Mixture Model (GMM) language recognition system. The GMM system setup and description are given in [9]. Briefly, each language model consisted of a GMM with 2048 mixture components. SDC features were extracted using the parameter specification 7-1-3-7; the features were post-processed using the feature mapping technique [10]. Language models were gender dependent, so a total of 24 models were used for the 12 target languages.

We first considered the effect of the backend on language recognition performance for the SVM-only case, see Figure 6. In the figure, we compare the performance of three systems. As can be seen, the "raw" SVM scores (i.e., no backend normalization) perform considerably worse than a backend processed score. If we do only LLR normalization as in (8) on the SVM scores, this performs substantially better. Finally, using the full backend process described in Section 5 performs the best.

We next considered the performance of the system relative to a GMM language recognition system, see Figure 7. In the figure, we see that the new SVM system is performing competitively with the state-of-the-art GMM system. The figure also shows the fusion of the two systems. Fusion was accomplished with the backend system discussed in Section 5. As the figure illustrates, the fusion combination works extremely well, significantly outperforming both individual systems. The equal error rates for these different systems is shown in Table 1.

Finally, we performed several additional experiments to try to tune the system. We tried gender dependent models. We also used a full covariance matrix in the GLDS kernel in (6) with an SDC parameterization of 7-1-3-4. Neither of these experiments improved performance.

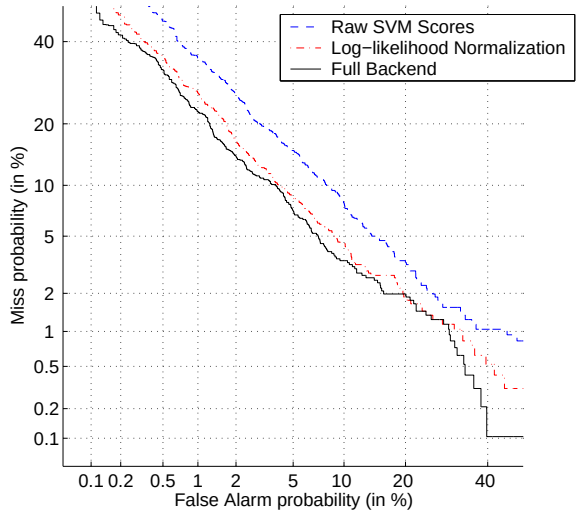

Figure 6: Comparison of different strategies for the backend for SVM-only language recognition

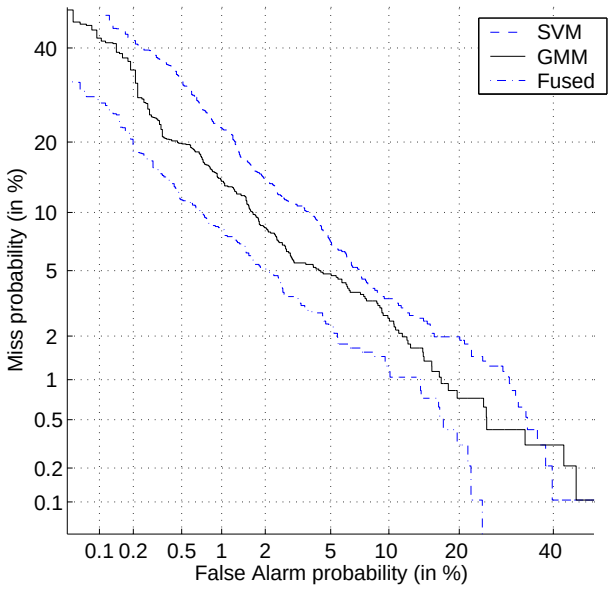

Figure 7: Performance of three different systems on the NIST 2003 language recognition evaluation for 30s duration tests

Table 1: EER performance of the systems for the 30s test

System	EER
SVM	6.1%
GMM	4.8%
Fused	3.2%

7. Conclusion

We have presented a new language recognition system based upon support vector machines. This new approach was based upon shifted delta cepstral coefficient and a novel sequence kernel. Comparison with another purely acoustic approach based upon the GMM showed competitive performance. Fusing the SVM and GMM system yielded further substantial gains in accuracy. Overall, we have demonstrated the interesting potential of the new approach. A fertile area for future work is tuning the SDC features and backend methods to the SVM approach.

8. References

[1] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[2] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *International Conference on Spoken Language Processing*, 2002, pp. 89–92.

[3] E. Wong, J. Pelecanos, S. Myers, and S. Sridharan, "Language identification using efficient gaussian mixture model analysis," in *Australian International Conference on Speech Science and Technology*, 2000.

[4] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2002, pp. 161–164.

[5] "The 2003 NIST language recognition evaluation plan," http://www.nist.gov/speech/tests/lang/index.htm, 2003.

[6] Alvin F. Martin and Mark A. Przybocki, "NIST 2003 language recognition evaluation," in *Proceedings of Eurospeech*, 2003, pp. 1341–1344.

[7] Nello Cristianini and John Shawe-Taylor, *Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[8] Ronan Collobert and Samy Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[9] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic, and discriminative approaches to automatic language identification," in *Proceedings of Eurospeech*, 2003, pp. 1345–1348.

[10] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2003, vol. 2, pp. II–53–56.