THE ADAPTATION SCHEMES IN PR-SVM BASED LANGUAGE RECOGNITION

Xu Bing, Song Yan, Dai LiRong

iFlytek Speech Lab, University of Science and Technology of China, Hefei
binghsu@ustc.edu, {songy, lrdai}@ustc.edu.cn

ABSTRACT

Phonetic-based systems usually convert the input speech into token (i.e. word, phone etc.) sequence and determine the target language from the statistics of the token sequences on different languages. Generally, there are two kinds of statistical representation for token sequences, N-gram language model (PR-LM) and support vector machines (PR-SVM) to perform language classification. In this paper we focus on PR-SVM method. One problem of the PR-SVM is that the statistical representation based on utterance is sparse and inaccurate. To tackle this issue, the adaptation schemes in PR-SVM framework are proposed in this paper. There are two schemes to be used: 1) Adaptation from the Universal N-gram Language Model (UNLM) trained on all languages; 2) Adaptation from the Low-Order N-gram Language Model (LONLM). The experimental results on 2007 NIST LRE tasks show that our method achieves significant gains over the unadapted model.

Index Terms— Language Recognition, Support Vector Machine, Adaptation

1. INTRODUCTION

Since the phone recognizer followed by language models (PRLM) [1] was proposed for language recognition, the language modeling on phonotactic features has become a prevailing paradigm in language recognition.

Recently, several methods were proposed to improve the performance on phonetic-related methods. In [2, 3], Lattices from phone decoder were introduced to provide additional information, which can be weighted by the posterior of alternate hypothesis. In [4], a discriminative SVM-based language modeling method, PR-SVM, was proposed. In this method, sequence kernels are constructed by viewing a speech segment as a *document* of tokens. The feature vector for SVM is a scaling version of co-occurrence probabilities (or N-grams of token) in an utterance.

However, an utterance will not contain all potential N-grams, which results in a sparse feature vector. That is, only a small subset of entries in feature vectors is non-zero. Meanwhile, the discriminative phonotactic feature for specific language may be contained in different order N-

gram statistics. With the increase of the order, the number of unique N-grams grows exponentially. Thus, the statistics of high-order N-gram from just one utterance can not be accurate.

In this paper, the adaptation schemes in PR-SVM framework are proposed, which can effectively alleviate the aforementioned problems. There are two schemes in our implementation:

1) Adaptation from the LONLM, which is similar as back-off (smoothing) scheme in language modeling. In this scheme, the low-order N-gram probabilities of an utterance are considered to be more accurate, from which the corresponding high-order ones are adapted.

2) Adaptation from the UNLM, which is motivated by the GMM-SVM [8]. The UNLM was first estimated on tokens obtained from all training data. For each utterance, its N-gram probabilities are adapted from the UNLM.

To evaluate the effectiveness of these schemes, several experiments are conducted on 2007 NIST LRE evaluation tasks. The experimental results show that both of the adaptation schemes can significantly improve the system performance.

This paper is arranged as follows. Section 2 gives the overview of PR-SVM system. In section 3, the proposed adaptation schemes are detailed. In section 4, the experimental results on 2007 NIST LRE tasks are shown to prove the effectiveness of our proposed schemes, followed by the discussion and future work in section 5.

2. PR-SVM FRAMEWORK OVERVIEW

The flowchart of PR-SVM language recognition system is illustrated in Figure 1, which mainly consists of a frontend phone recognizer, a post-processing step and backend language models based on SVM. Given an utterance, the token sequence or lattice is firstly obtained as the output of the phone recognizer. Then the “bag-of-N-gram” feature vector is constructed for each utterance in the post-processing step. Then target SVM models are trained and used for classification. Details are given below.

2.1. Phone Recognizer

Figure 1 Flowchart of basic PR-SVM system

In this paper, the Hungarian phone recognizer released by Brno University was employed as frontend of our system [5], which comprises 62 3-state monophone HMMs with null grammar. Instead of traditional Mel Frequency Cepstral Coefficient (MFCC) feature, a temporal pattern feature is used to improve accuracy of telephony speech recognition. Then neural network is trained to estimate HMM state posterior probabilities.

2.2. SVM Classifier

A support vector machine is a two-class classifier constructed from sums of a kernel function $K(x, y) = b(x)' b(y)$, where $b(x)$ is a mapping from input space to a high dimensional space. The optimized SVM solution is

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

where $\sum_{i=1}^N \alpha_i = 0$, The vectors x_i are support vectors and obtained from the training set through an optimization process.

In our PR-SVM solution, training is performed with the one-versus-others strategy using package SVM-Torch [6].

2.3 Kernel Construction

The core of the SVM classifier is the construction of the kernel, which defines the similarity between two utterances. Following the work in [4], a “bag-of-N-grams” representation of an utterance is used.

Suppose that there are a phone inventory of M phones $W = \{w_1, w_2, \dots, w_M\}$ and a phone sequence $U = t_1 t_2 \dots t_n$ of an utterance, the probabilities of unique N-grams in U are calculated as follows

$$p(d_i | U) = \frac{\text{Count}(d_i | U)}{\sum_j \text{Count}(d_j | U)} \quad (2)$$

Here d_i is one of the unique N-grams for a fixed N. Take 2-gram for example, d_i is one of the following M^2 2-grams $\{w_1 w_1, w_1 w_2, \dots, w_M w_{M-1}, w_M w_M\}$.

In order to make sure that N-grams with large probabilities will not dominate the similarity in kernel, an

utterance-based normalization method is needed, which can be formulated as

$$p'(d_i | U) = D_i p(d_i | U) = \sqrt{\frac{1}{p(d_i | all)}} p(d_i | U) \quad (3)$$

Here $D_i = 1/p(d_i | all)$ is calculated from the probability across all training data. These weighted N-gram probabilities are used as entries of a high dimensional vector. Then a kernel between two utterances can be formed. For two sequences U and V , the kernel is

$$K(U, V) = \sum_i D_i^2 p(d_i | U) p(d_i | V) \quad (4)$$

2.4 Lattice Decoding

Lattices were first used by Gauvain et al in PRLM language recognition system [2]. By calculating N-gram statistics from lattices rather than from 1-best phone sequences, a more accurate estimation of N-gram probabilities can be obtained. In this case, the counts in (2) are replaced with expectations over all hypotheses in lattice L . This alternate expression for (2) is

$$p(d_i | L) = \frac{E(d_i | L)}{\sum_j E(d_j | L)} \quad (5)$$

Here

$$E(d_i | L) = \sum_{H \in L} p(H | L) \text{Count}(d_i | H) \quad (6)$$

H is a hypothesis in lattice L , i.e., a possible phone sequence. And $p(H | L)$ is its posterior probability.

Given the original speech X and the acoustic model (phone recognizer) Λ , the posterior probability can be calculated as follows.

$$p(H | L) = \frac{p(X | H, \Lambda)^\lambda}{\sum_{H' \in L} p(X | H', \Lambda)^\lambda} \quad (7)$$

Here $p(X | H, \Lambda)$ is acoustic model score for speech X , 1-best phone sequence is the hypothesis with the largest posterior probability (or acoustic model score). The role of λ is to vary the distribution of the posterior over H . The lattice output of phone recognizer is a natural extension of string output (1-best) by using HTK toolkit [7].

3. ADAPTATION SCHEMES

In GMM-SVM [8], the GMM parameters of an utterance are obtained by updating a well trained Universal Background Model (UBM) model, and the supervector is constructed by stacking the mean parameters of the adapted mixture components. Similarly, the “bag-of-n-gram” feature vector in PR-SVM can be viewed as the N-gram language model parameters of an utterance, which models the high-level phonotactic context information.

Figure 2 Our Proposed Post-Processing Scheme

However, “bag-of-n-gram” feature vector is in fact a scaling version of co-occurrence probabilities, which is sparse and inaccurate for an utterance with limited duration.

To further improve the performance of the PR-SVM, we proposed two adaptation schemes in the following sections. The motivation is to obtain an accurate representation of an utterance by adapting from accurate models, i.e. the LONLM or UNLM etc., as shown in Figure 2.

3.1 Adaptation from LONLM

In the traditional construction of N-gram statistic language model, various discounting and smoothing approaches have been used to deal with unseen N-grams in training data. It is assumed that with the finite training material, the low-order statistics are more accurate than the high-order ones.

In the case of lattice output, we first calculate low-order probabilities as equation (5). Then the high-order N-gram probabilities were adapted from the low-order probabilities by simple interpolation. For example, the bigram statistics is interpolated from the unigram statistics as

$$\hat{p}(w_i w_j | L) = \frac{\alpha}{M} [p(w_i | L) + p(w_j | L)] + (1 - \alpha) p(w_i w_j | L) \quad (8)$$

Here $\hat{p}(w_i w_j | L)$ is the adapted 2-gram probability in lattice L , $\alpha \in [0, 0.5]$ is the smoothing factor, $p(w_i | L)$, $p(w_j | L)$ and $p(w_i w_j | L)$ are original probabilities calculated from (5). M is the phone inventory size, which acts as a normalization factor to keep $\sum_{i,j=1 \dots M} \hat{p}(w_i w_j | L) = 1$. This process is repeated until N is reached.

3.2 Adaptation from UNLM

Motivated by the GMM-SVM system in [8], the second approach is to make use of the universal distribution of N-grams. An adaptation from the universal N-gram model is applied through linear interpolation. In the case of the lattice outputs, the interpolation is as follows

$$\hat{p}(d_i | L) = \beta p(d_i | all) + (1 - \beta) p(d_i | L) \quad (9)$$

The UNLM $p(d_i | all)$ is calculated from lattices across all training data. Note that the same probability is used in the weighting step (3). $\beta \in [0, 1]$ is the interpolation parameter.

4. EXPERIMENTS AND RESULTS

4.1 Experiment Setup

Experiments were conducted on 2007 NIST LRE tasks. The full evaluation data set consists of 7530 telephone utterances spanning the 30, 10, 3 second durations (2510 each). Our system was evaluated under the closed set condition which limits the evaluation data to 6474 utterances (2158 for each duration) in 14 languages (Arabic, Bengali, Chinese, English, Farsi, German, Hindustani, Japanese, Korean, Russian, Spanish, Tamil, Thai, Vietnamese).

Training data set comprises CallFriend corpus containing telephone speech of 15 different languages or dialects (each contains 20 2-channel telephone conversations) and 2007 NIST LRE training Data of 8 different language or dialects (each contains 40 10-minute single-channel telephone conversations) and about 13 hours of Russian from the 2004 and 2006 NIST SRE corpus.

Development data set comes from 2003 NIST LRE evaluation data and random select of segmented 2007 NIST LRE training data.

The training data was segmented to around 30 second using an energy-based speech activity detector. After removal of non-speech data using the same speech activity detector, lattices were generated across all the above data using the phone recognizer described in Section 2.1. N-gram expectations in (6) were calculated using an empirical λ on development data.

We used a reduced phone set obtained by merging short/long variants, which makes it possible to calculate the high-order N-grams.

4.2 Results

Table 1 compares the equal error rates (EER) of 1-best and lattice decoding on 2007 NIST LRE with the basic 3-gram PR-SVM system. It shows that use of lattices outperforms significantly over the 1-best output.

Table 2 summarizes the experimental results using different adaptation approaches on 2007 NIST LRE under the 3-gram PR-SVM-lattice architecture. Empirical values of α and β on development data were used. We can see that back-off adaptation from the low-order N-gram gives 15%, 18% and 16% improvements respectively for 30, 10, 3 second trials, and the corresponding improvements brought by UNLM adaptation are 13%, 19% and 16%.

Table 2 also gives the fusion of these two adaptation approaches. Through simple linear fusion, we achieved a great reduction of 28% in EER from the baseline on 30 second trials. But on 10 and 3 second duration, we did not see as much improvement.

Table 1 Comparison of 1-best and lattice decoding in EER(%)

decoding	30sec	10sec	3sec
1-best	7.0	17.9	32.2
lattice	3.9	12.0	26.7

Table 2 Comparison of EERs(%) for different adaptation approaches

	30sec	10sec	3sec
Baseline	3.9	12.0	26.7
LONLM Adaptation	3.3	9.8	22.5
UNLM Adaptation	3.4	9.7	22.3
Fusion	2.8	9.5	22.4

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed the adaptation schemes in PR-SVM system for language recognition, which can effectively alleviate the sparseness and inaccuracy of the statistical representation of the utterance. The experimental results show advantage of our proposed schemes. The future work will be as follows. 1. More flexible adaptation schemes need to be found; 2. Now we are constrained to 3-gram since the dimension grows exponentially as N increases, recent works of [9, 10] make it possible to reach higher order N-grams. And intuitively, adaptation may work even better with higher order.

6. 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] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in Proceedings of ICSLP, 2004.
- [3] W.M. Campbell, F. Richardson, D.A. Reynolds, "Language Recognition with Word Lattices and Support Vector Machines," in ICCASP, 2007.
- [4] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "High-level speaker verification with support vector machines," in Proceedings of ICASSP, 2004, pp. 1–73–76.
- [5] Petr Schwarz, Matejka Pavel, and Jan Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in Proceedings of ICASSP, 2006, pp. 325–328.
- [6] 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.
- [7] S. Young, Gunnar Evermann, Thomas Hain, D. Kershaw, Gareth Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, The HTK book, Entropic, Ltd., Cambridge, UK, 2002.
- [8] W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff, "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation," in Proceedings of ICASSP, 2006.
- [9] Bin Ma, Rong Tong, Haizhou Li, "Discriminative Vector for Spoken Language Recognition", in ICCASP, 2007.
- [10] F.S. Richardson, W.M. Campbell, "Language recognition with discriminative keyword selection," in ICCASP, 2008