Hidden Logistic Linear Regression for Support Vector Machine based Phone Verification

Bo Li, Khe Chai Sim
School of Computing, National University of Singapore.
1 Computing, 13 Computing Drive, Singapore 117417
\{li-bo, simkc\}@comp.nus.edu.sg

Abstract

Phone verification approach to mispronunciation detection using a combination of Neural Network (NN) and Support Vector Machine (SVM) has been shown to yield improved verification performance. This approach uses a NN to predict the HMM state posterior probabilities. The average posterior probability vectors computed over each phone segment are used as input features to a SVM back-end to generate the final verification scores. In this paper, a novel Hidden Logistic Feature (HLF) for SVM back-end is proposed, where the sigmoid activations from the hidden layer that contain rich information of the NN is used instead of the output layer and the generation of HLFs can be interpreted as a Hidden Logistic Linear Regression process. Experiments on the TIMIT database show that the proposed HLF gives the lowest Equal Error Rate of 3.63%.

Index Terms: Phone verification, neural network, support vector machine, logistic linear regression

1. Introduction

Speech verification is generally treated as a hypothesis testing problem or classification problem [1]. For phoneme verification, each phone to be verified has a verification score generated by the system and a decision threshold is applied to the verification scores, above which the speech segment corresponds to this phone and vice versa. Typically, the phone posterior probabilities computed from a phone recogniser are adopted as the verification scores. Instead of the conventional Hidden Markov Models (HMMs), the hybrid of Neural Network (NN) and HMM (NN/HMM) system has been found to yield superior phone recognition [2] and phone verification [3] performance. Support Vector Machine (SVM) can also be incorporated as a back-end processing to enhance the verification performance.

There have been several approaches of integrating SVM into verification tasks. The Fisher kernel [4] maps the resulting score-sequences of the Gaussian Mixture Models (GMMs) into a high dimensional score-space where a SVM is then used to classify the data in a second step [5]. For phone verification, instead of directly using the target phone’s posterior probabilities alone, the entire vector of posterior probabilities computed for all the phones can be used as the input features to the SVM back-end. Using state posterior probabilities generated by a NN system has been found to yield better performance compared to phone posterior probabilities generated by HMMs for SVM-based phone verification [3].

This paper further investigated the activations from the hidden layer neurons of the NN as the input features to SVM back-end. In the NN structure, the information is refined and also reduced layer by layer. The hidden layer always contains additional discriminative information than the output. Normalisation is usually applied at the output layer to make the summation of all the outputs to 1, leading to much information loss. The output layer is compulsory in classification for the class label prediction, but unnecessary in verification due to the second stage SVM classifiers, which are quiet efficient at handling high dimensional features. To maintain as much information as possible, the high dimension hidden outputs with rich classification information instead of the low dimension posterior features are preferable for the SVM back-end. From another perspective, the NN’s input to hidden part can be viewed as a non-linear SVM kernel transforming the acoustic features into high-dimensional feature vectors which are suitable for the subsequent SVM classification. Since the hidden layer uses the sigmoid function, each hidden unit can be viewed as performing a binomial Logistic Linear Regression (LLR) on some hidden attributes which are then projected onto the state posterior probability space in the final layer of the NN. This method is thus referred to as Hidden Logistic Linear Regression (HLLR) and the hidden activations are named as the Hidden Logistic Features (HLFs).

The paper is organised as follows: Section 2 describes the phone verification framework. Section 3 explains the NN/HMM hybrid phone recognition system, followed by the discussion of the HLLR and HLF in Section 4. Experimental results are reported on the TIMIT database in Section 5.

2. System Description

Figure 1: Comparison between Posterior Feature (PF) and Hidden Logistic Feature (HLF).

The same phone verification architecture as [3] is employed in this work. Given the speech waveform and its corresponding word-level orthographic transcription, standard MFCC acoustic features are extracted and forwarded to the NN/HMM acoustic models for recognition. With acoustic models’ state level posterior probabilities, the phone segmentation can be achieved by combining the adjacent phone states for the same phone. Veri-
fication features are extracted to represent each phone segment. These features are then fed into a SVM back-end classifier to produce the final verification scores. In [3], state-level Average Posterior Feature (SAF) PF yields much better verification performance than phone-level Average Posterior Feature (PAF). In this work, another state level features extracted from the NN/HMM system - HLF generated by the HLLR process - is investigated (see Figure 1). In this paper, we simply refer to the SAF in [3] as Posterior Feature (PF).

3. NN/HMM Hybrid Phone Recogniser

In this work, the NN/HMM hybrid system [6] was adopted for phone verification. It was shown in [3] that the NN/HMM systems outperformed the conventional HMM systems in generating high quality posterior probabilities for phone verification. Two different structures of NN for monophone recognition are investigated: (i) the fully connected three-layer Feedforward Neural Network (FNN) (Figure 1); (ii) the Modular Neural Network (MNN) (Figure 2).

Standard Error Back-Propagation (EBP) method was employed for all the NN training and the activation functions are all the sigmoid function for hidden layers and the softmax function for output layers. Experience with HNM technology has shown that context-dependent phonetic models improves recognition accuracy significantly [8]. Thus the feature vector $f_t$, at time $t$, used for training is obtained by stacking multiple observation vectors around time $t$ instead of using only the $o_t$, i.e.

$$f_t = [o_{t-w} \ldots o_t^T \ldots o_{t+w}^T]^T,$$

where $w$ is the number of preceding and succeeding observation vectors to be included in $f_t$ to capture longer temporal information. The effective window length is therefore $2w + 1$.

In the time-varying speech signal, the pre- and post-frame contexts may have different effects on the current one. We thus experimented with the Modular Neural Network (MNN) as illustrated in Figure 2, which splits the context information into two independent modules, Left Context Neural Network (LC-NN) and Right Context Neural Network (RC-NN), and an Intermediary Neural Network (I-NN). Each of them is a FNN. The LC-NN and RC-NN operate separately on its corresponding context and the central frame, i.e. the feature vector for LC-NN, $f_{LT}$, and the feature vector for RC-NN, $f_{RT}$ are:

$$f_{LT} = [o_{t-w}^T \ldots o_{t-(w-1)}^T \ldots o_t^T]^T,$$

The I-NN concatenates the output of individual context NN at frame level as the input feature vector and predicates the final state level posterior probabilities. This MNN structure yields better recognition results compared to FNN trained on acoustic features covering the same temporal contexts.

Similar to [3], segment level posterior, namely the Average Posterior Feature (APF), is computed as:

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} p_i,$$

where $n$ is the number of frames aligned to the phone segment and $p_i$ is the posterior feature vector of the $i$th frame. Given the set of $\hat{p}$ and the corresponding target phones, SVM classifiers are trained, one for each phone, in a one-versus-rest manner. In [3], state-level average posterior features were found to yield superior performance compared with phone-level average posterior features as they contain more information, such as FNN’s PF (Figure 1) and MNN’s LC-PF, RC-PF and I-PF (Figure 2).

This paper further investigated high dimensional HLFs to take the advantage of NN’s hidden layer activations which contains more discriminative information. This HLF’s generation process is referred to as the Hidden Logistic Linear Regression (HLLR) and will be introduced in the following section.

4. Hidden Logistic Linear Regression

Logistic Linear Regression (LLR) is a generalised linear model used for binomial regression to predict the probability of occurrence of an event by fitting data to a logistic curve [9]. It is commonly used for regression analysis on the posterior probabilities. The logistic function is the same as the sigmoid function used in the NN’s hidden layer.

In the hybrid NN/HMM system, NN acoustic models are trained to optimise the state posterior prediction. The generated posterior features are normalised to sum to one, which greatly eliminates the information contained in the PFs and make them unsuitable for verification. It would be useful to investigate the hidden layer features that generated the PFs for phone verification. Due to the sigmoid (binomial logistic) function used in the NN’s hidden layer, the output from each hidden component can be regarded as the Hidden Logistic Feature (HLF) of a latent attribute and as far as the hidden layer is concerned, the NN training can be viewed as performing a Logistic Linear Regression (LLR) on these latent attributes. This is referred to as Hidden LLR (HLLR) which is effectively a process of transforming acoustic features into high dimensional posterior probabilities of the latent attributes. One may also view HLLR as a form of nonlinear SVM kernel which expands the acoustic features into a high dimensional feature space. The number of latent attributes (size of the hidden layer of the NN) can be adjusted to control the complexity of the HLLR model. Typically, hundreds or thousands of hidden units are used to yield good performance. Therefore, HLLR forms an array of independent LLR for each latent attribute. Note that, unlike the standard LLR, the target values for HLLR are not explicitly defined. In fact, the latent attributes do not correspond to specific physical meaning. They are determined implicitly through the standard NN training such that the hidden layer values will yield the optimum prediction of the final HMM state posterior probabilities. Hence, it is known as hidden LLR.
5. Experimental Results

This section presents the experimental results of phone verification on the TIMIT database. A summary of the training and testing data sets used in this work is tabulated in Table 1. All the phoneme recognisers are monophone systems. The reduced TIMIT phone set with 39 phones (including a pause model) was used. NNs were trained using the 39 dimensional MFCC feature vectors that consists of 13 static coefficients (12 MFCC plus the C0 energy term) and the Δ and ΔΔ parameters. Feature vectors are extracted at every 10 milliseconds. Each frame is modelled with a 3-state HMM with a left-to-right topology. The QuickNet\(^1\) software package was used for neural network training and SVM\(^2\)light\(^2\) was used to train the SVM models for back-end verification.

Table 1: Summary of the TIMIT training and testing data sets used in this work.

<table>
<thead>
<tr>
<th>Data Set</th>
<th># of Speakers</th>
<th># of Utterances</th>
<th>Amount of Data (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>492</td>
<td>3696</td>
<td>3.12</td>
</tr>
<tr>
<td>test</td>
<td>168</td>
<td>1344</td>
<td>1.14</td>
</tr>
</tbody>
</table>

5.1. Phone Recognition

A good phone recogniser should be adopted for high quality PFs and HLFs generation. We used NN’s Frame Error Rate (FER) and HMM’s Phone Error Rate (PER) for performance evaluation. Both FNN and MNN are trained on the same feature vector \(f_t\) (Eq.(1) with \(w = 5\)). The number of hidden units of FNN varies from 1000 to 2000 by 100 for an optimum setting and the FERs and PERs are illustrated in Figure 3. Using 1500 hidden units in FNN led to the best phoneme recognition performance with 34.18% FER and 34.24% PER.

![Figure 3: Trends of FER and PER results for FNNs with the number of hidden units varying from 1000 to 2000.](http://example.com/figure3)

The MNN system has the same input as FNN. But each context NN only takes 6 out of the 11 frames as the input feature vector (Eq.(2) and Eq.(3) with \(w = 5\)). The MNN with approximately the same computational cost as the FNN is more interesting to investigate, thus we did not tune the number of hidden units for sub-NNs in MNN. 750 hidden units are directly employed for both LC-NN and RC-NN, giving a total of 1500 hidden units, which has the same dimension as the FNN’s hidden layer. The Intermediary NN (I-NN) has 500 hidden units, which yields the best MNN phone recognition performance given LC-NN and RC-NN’s outputs. Detailed recognition performance for MNN are listed in Table 2. The FERs and PERs of LC-NN and RC-NN are higher than the FNN because they comprised only 750 hidden units and was trained on partial context information. The whole MNN system gives a significantly improved PER performance of 32.30% (i.e., the performance of I-NN), 1.94% absolute error reduction from FNN system, because of the increased structure complexity of MNN.

### Table 2: FER(%) and PER(%) for phoneme recognition using MNN.

<table>
<thead>
<tr>
<th>Modulars</th>
<th>FER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC-NN</td>
<td>36.78</td>
<td>36.74</td>
</tr>
<tr>
<td>RC-NN</td>
<td>36.91</td>
<td>36.81</td>
</tr>
<tr>
<td>I-NN</td>
<td>31.42</td>
<td>32.30</td>
</tr>
</tbody>
</table>

### Table 3: Comparison of EER(%) performance of using different Frontal Features and Hidden Logistic Features with linear SVM kernel.

<table>
<thead>
<tr>
<th>Systems</th>
<th>Features</th>
<th>Dimension</th>
<th>EER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFs</td>
<td>FNN</td>
<td>117</td>
<td>4.28</td>
</tr>
<tr>
<td>MNN</td>
<td>LC-PF</td>
<td>117</td>
<td>4.41</td>
</tr>
<tr>
<td></td>
<td>RC-PF</td>
<td>117</td>
<td>4.58</td>
</tr>
<tr>
<td></td>
<td>I-PF</td>
<td>117</td>
<td>5.24</td>
</tr>
<tr>
<td>HLFs</td>
<td>FNN</td>
<td>1500</td>
<td>4.13</td>
</tr>
<tr>
<td>MNN</td>
<td>LC-RLF</td>
<td>750</td>
<td>4.38</td>
</tr>
<tr>
<td></td>
<td>RC-RLF</td>
<td>750</td>
<td>4.44</td>
</tr>
<tr>
<td></td>
<td>I-RLF</td>
<td>500</td>
<td>4.09</td>
</tr>
</tbody>
</table>

In recognition, we do not care the accuracy for each phone. Even some infrequently used phones are completely wrongly recognised, the overall recognition accuracy may not be affected too much. While in verification, the EERs are accumulated. To get a lower EER, each of them should have a lower EER. For example, in our experiments, there are totally 45821 test phone segments; 7.48% of them are phone segment “p” and only 0.37% of them are phone segment “uh”. 108 “p”s are mis-
5.3. Phone Verification With HLFs

Instead of the normalised PFs, HLFs before projected to the lower dimension PFs carry more discriminative information and probably be more suitable for verification. Experimental results on both FNN and MNN using HLFs are shown in Table 3.

Compared with the PF results, the HLFs consistently outperformed PFs. For different systems with HLF features, MNN’s LC-HLF and RC-HLF are still worse than FNN’s HLF due to the lower dimension of hidden space and the partial context information involved in training. MNN’s 1-HLF gives 4.09% EER, 0.04% absolute error reduction from FNN’s HLF, which shows that the HLF features are more stable for phone verification. Besides, from Table 3, higher dimension (1500 and 750) does not guarantee a better performance, the information captured in the features is much more important.

5.4. Combined Full Context Features

From the results of PFs and HLFs (Table 3), we observed that the full context features, such as NN’s PF and HLF and MNN’s 1-PF and I-HLF, consistently outperformed the partial context features like MNN’s LC-PF and LC-HLF. We combined the partial context features of the existing systems for SVM back-end verification. The combination is a frame-level direct concatenation, which leads the dimensionality of “LC-PF & RC-PF” and “LC-HLF & RC-HLF” to be 234 and 1500, respectively. The verification performance is illustrated in Table 4.

Table 4: Comparison of EER(%) performance of using separate and combined PF and HLF of MNN system with linear SVM kernel.

<table>
<thead>
<tr>
<th>Features</th>
<th>Dimension</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC-PF</td>
<td>117</td>
<td>4.41</td>
</tr>
<tr>
<td>RC-PF</td>
<td>117</td>
<td>4.58</td>
</tr>
<tr>
<td>LC-HLF</td>
<td>750</td>
<td>4.38</td>
</tr>
<tr>
<td>RC-HLF</td>
<td>750</td>
<td>4.44</td>
</tr>
<tr>
<td>LC-PF &amp; RC-PF</td>
<td>234</td>
<td>4.38</td>
</tr>
<tr>
<td>LC-HLF &amp; RC-HLF</td>
<td>1500</td>
<td>3.88</td>
</tr>
</tbody>
</table>

With the combined full context features, we achieved absolute error reduction of 0.03% and 0.50% over the partial context features. However, the “LC-PF & RC-PF” still does not perform better than the basic FNN system’s PF, which shows that the normalised posterior features are not suitable for concatenation. In both LC-NN and RC-NN, the PFs are trained on the same target values, i.e. both PFs represent the same posterior probabilities. On the other hand, HLFs are derived from different context features for accurate posterior prediction, thus contain much information and more suitable for verification.

The combined “LC-HLF & RC-HLF” yields the superior verification performance, 3.88% EER achieving absolute EER reductions of 0.25% from FNN’s HLF (Table 3) with the same dimensionality. It also gives us the advantage of lower computational cost. Instead of training a FNN of approximately 75 million ((11*39)*1500 + 117) weight parameters, for the two context NNs, about 40 million ((6*39)*750 + 117*2) weights to be estimated. Meanwhile, the two context NN are independent to each other, parallel training is applicable. In our experiments, the FNN training takes about 6 hours, while only about 2 hours for each context NN training. Thus the “LC-HLF & RC-HLF” not only yields the best performance but also has a comparable or even lower computational cost.

Finally, two more complex kernels are adopted for further improvement (Table 5) and the Radial Basis Function (RBF) kernel yields the lowest EER of 3.63%, which is a further 0.25% absolute EER reduction compared to using a linear kernel.

6. Conclusions

This paper has proposed the Hidden Logistic Feature (HLF) generated by a Hidden Logistic Linear Regression (HLLR) process for SVM-based phone verification. The acoustic features are transformed into high dimensional feature vectors using the hidden layer activations from a three-layer Neural Network that was trained to predict the HMM state posterior probabilities. Experimental results on TIMIT show that HLFs consistently outperformed the NN output Posterior Features (PFs). Moreover, further performance improvement can be achieved by combining the HLFs from NNs trained on acoustic features that are split into the left and right contexts. The lowest EER of 3.63% was obtained using HLFs from the combination of Left Context Neural Network (LC-NN) and Right Context Neural Network (RC-NN) with the radial basis function kernel.

7. Acknowledgements

This research is done for CSIDM Project No.CSIDM-200806 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

8. References