SOFT MARGIN ESTIMATION OF GAUSSIAN MIXTURE MODEL PARAMETERS FOR SPOKEN LANGUAGE RECOGNITION

Donglai Zhu1, Bin Ma1, Haizhou Li1,2

1Institute for Infocomm Research, Singapore
2Department of Computer Science and Statistics, University of Eastern Finland, FI-80101 Joensuu, Finland
{dzhu,mabin,hli}@i2r.a-star.edu.sg

ABSTRACT

This paper extends our previous work on large margin estimation (LME) of GMM parameters with extend Baum-Welch (EBW) for spoken language recognition. To overcome the problem in the LME that negative samples in the training set are not used in parameter estimation, we propose a soft margin estimation (SME) method in this paper. The soft margin is scaled by a loss function measuring the distance between a negative sample and the classification boundary. We formulate the constrained optimization of SME as an unconstrained optimization among both positive samples and negative samples using a penalty function, and update the GMM parameters with the EBW algorithm. Experiments on the NIST language recognition evaluation (LRE) 2007 task show that the SME method effectively improves the LME performance.

Index Terms: spoken language recognition, soft margin estimation, extended Baum-Welch

1. INTRODUCTION

Discriminative training (DT) of acoustic modeling has been proved effective for spoken language recognition (SLR). The maximum mutual information (MMI) estimation of Gaussian mixture model (GMM) in general outperforms the maximum likelihood (ML) estimation of GMM [1]. The support vector machine (SVM) is also successfully used, e.g. the GMM super-vector (GSV) method in which Kullback-Leibler (KL) divergence between GMMs is modeled by SVM, and the pushing model that pushes the SVM training parameters back to the GMM model for scoring [2][3]. Recently we have proposed a large margin estimation (LME) method to train GMM parameters [4]. While the traditional DT methods such as MMI and minimum classification error (MCE) [5] attempt to minimize empirical error rates on training data, the LME method attempts to maximize the multi-class separation margin so as to improve the generalization ability to better classify unseen test data. A problem in LME is that the meaningful margin is defined on correctly classified samples (the so-called positive samples) in training set. If the training data are inseparable, the mis-classified samples (the so-called negative samples) are not used for parameter estimation.

To deal with the inseparable data, the SVM either uses a kernel function to map training samples into a nonlinear feature space which has a separable hyperplane, or defines a soft margin which is relaxed to permit margin violation by negative samples [6]. Recent studies have incorporated the soft margin concept into the HMM framework. The soft margin estimation (SME) approach in [7] directly uses the soft margin concept to define the objective function as an approximate test risk bound. The maximum relative margin estimation (MRME) and soft-LME approaches optimize the objective functions which linearly combine the minimum margin among positive samples and an average error function among negative samples [8][9]. However, these approaches deal with the closest competing models only, and it is hard (or certain assumptions are required) to formulate their objective functions as convex optimization problems. An LME approach in [10] attempts to yield a convex optimization of margins which are augmented with soft-max for multiple Gaussians and with Hamming distance for the HMM state sequence. The Hamming distance margin is a case of the loss-scaled soft margin [11], which has the advantage that the inference of model parameter estimation can be incorporated into the extended Baum-Welch (EBW) algorithm. For example, the Hamming distance margin is adopted in a penalty-based LME method [12], and the margin scale is defined as the number of recognition errors in Boosted-MMI [13].

This paper is focused on the training of language-dependent GMMs in the GMM-UBM framework. We extend the LME method in our previous work [4] to SME using a loss-scaled margin. Because SLR is not a sequential classification problem that the Hamming distance scale is designed for, we define the margin scale as a hinge function of the log likelihood ratio between the target language and the competing languages. By using a penalty function to penalize the constraint violations incurred by negative samples, we derive an integrated objective function among both positive and negative samples, which can be effectively optimized using the EBW algorithm. Our method differs from [12] and [13] in two aspects: the margin scale in our method is likelihood ratio instead of error counts, and the target language of training samples is excluded from the denominator accumulation in the inference of parameter estimation. Experiments show that SME outperforms LME on the NIST LRE 2007 task.

The paper is organized as follows. Section 2 describes the concept of soft margin estimation of GMM. Section 3 presents derivation of GMM update formulae with EBW. Section 4 presents experimental setup and results. Conclusions are drawn in section 5.

2. SOFT MARGIN GMM

Let’s assume there are $L$ target languages to be recognized. The training data set consists of a collection of speech segments $\mathcal{X} = \{X_s, s = 1, \ldots, S\}$ where each segment is a sequence of feature vectors $X_s = \{x_{st}, t = 1, \ldots, T_s\}$ and $x_{st} = \{x_{st1}, \ldots, x_{std}\}$ are $D$-dimensional feature vectors. Each speech segment is labeled with one of languages denoted as $L = \{l_s, 1 \leq l_s \leq L, s = 1, \ldots, S\}$. Each language is modeled with a Gaussian mixture model of which parameters are denoted as $\lambda_l = \{c_{lm},$
\( \mu_m, \Sigma_m, m = 1, \ldots, M \) \}, where \( M \) is the number of Gaussian components, \( \mu_m = [\mu_{m1}, \ldots, \mu_{mD}]^T \) are \( D \)-dimensional mean vectors, and \( \Sigma_m = \text{diag}(\sigma_{m1}^2, \ldots, \sigma_{mD}^2) \) are diagonal covariance matrices. The parameter set is denoted by \( \Lambda = \{ \lambda_i, l = 1, \ldots, L \} \).

In our previous work [4], we proposed an LME method of GMM parameters. Denoting \( F(X_s, \lambda_i) \) as the discriminant function of \( X_s \) and a language model \( \lambda_i \), the multi-class separation margin for \( X_s \) is defined as

\[
d(X_s) = F(X_s, \lambda_i) - \max_{l \neq i} F(X_s, \lambda_l) .
\] (1)

If \( d(X_s) \leq 0 \), \( X_s \) will be incorrectly recognized by the GMM set \( \Lambda \); if \( d(X_s) > 0 \), \( X_s \) will be correctly recognized by the GMM set \( \Lambda \). A set of segments that are relatively close to the classification boundary in the right decision regions is defined as a support vector set:

\[
\Omega = \{ X_s | X_s \in \mathcal{X}, 0 \leq d(X_s) \leq \epsilon \} ,
\] (2)

where \( \epsilon > 0 \) is a positive number. Each segment in \( \Omega \) is called a support sample. The LM training is to make all support samples in training data, which is proved effective to improve the MMI performance in the LME training [4]. For \( \Omega \), the constraints in (4) are always satisfied so that the objective function becomes that of LME in (3). On the other hand, for the set of negative samples \( \mathcal{E} = \{ X_s | X_s \in \mathcal{X}, d(X_s) < 0 \} \), the zero bound in the hinge function in (6) is never reached. Combining (3) among positive samples and (6) among negative samples, we get the objective function without the hinge function as follows

\[
G = \rho - \frac{1}{\zeta} \sum_{s=1}^{S} \left( F(X_s, \lambda_{i_s}) - \max_{l \neq i_s} (F(X_s, \lambda_l) + \rho \delta_{il}) \right) , \] (6)

To convert (7) to a differentiable function, we replace the maximum with a soft-max upper bound as follows:

\[
G = \rho - \frac{1}{\eta} \sum_{s=1}^{S} \left( F(X_s, \lambda_{i_s}) - \frac{1}{\eta} \log \sum_{l \neq i_s} e^{F(X_s, \lambda_l) + \rho \delta_{il}} \right) , \] (8)

where \( \eta \geq 1 \). When \( \eta \to \infty \), the soft-max will approach the maximum. Let’s define

\[
F(X_s, \lambda_i) = \log p(X_s | \lambda_i)^{K_s} ,
\] (9)

where \( 0 < K_s < 1 \) is a scaling factor to increase the confusion between correct and competing languages [15]. Then (8) is rewritten as

\[
G = \rho + \frac{1}{\zeta} \sum_{s=1}^{S} \log \frac{p(X_s | \lambda_{i_s})^{K_s}}{\sum_{l \neq i_s} p(X_s | \lambda_l)^{K_s} e^{\rho \delta_{il}} \eta} . \] (10)

There are two major differences between (10) and the objective functions in the penalty-based LME [12] and the Boosted MMI [13]. First, the margin scale in our method is defined as a hinge function of log likelihood ratio in (5), while it is the Hamming distance of HMM state sequence in [12] and is the number of recognition errors in [13]. Second, the likelihood of correct class is excluded from the soft-max in (10) but is included in that of [12] and [13].

### 3. PARAMETER ESTIMATION

The parameters to be estimated in (10) include \( \rho \) and \( \Lambda \). It is discussed in [7] that there is little difference theoretically between two estimation methods: 1) to jointly estimate \( \rho \) and \( \Lambda \), and 2) to preset \( \rho \) in advance and estimate \( \Lambda \). We adopt the second method and write the objective function (10) with respect to \( \Lambda \) as follows

\[
G(\Lambda) = G_{\text{num}}^{\text{num}} - G_{\text{pen}}^{\text{pen}} = \sum_{s=1}^{S} \log p(X_s | \lambda_{i_s})^{K_s} - \sum_{s=1}^{S} \frac{1}{\eta} \log \sum_{l \neq i_s} p(X_s | \lambda_l)^{K_s} e^{\rho \delta_{il}} \eta . \] (11)

4991
Eq. (11) can be effectively optimized using the EBW algorithm, where a weak-sense auxiliary function for (10) is defined as [15]

$$Q(\Lambda|\Lambda') = Q_{\text{num}}(\Lambda|\Lambda') - Q_{\text{num}}^{(t)}(\Lambda|\Lambda') + Q_{\text{den}}^{(t)}(\Lambda) + \log p(\Lambda) .$$

(12)

$Q_{\text{num}}(\Lambda|\Lambda')$ and $Q_{\text{den}}^{(t)}(\Lambda)$ are auxiliary functions for $G_{\text{num}}(\Lambda)$ and $G_{\text{den}}^{(t)}(\Lambda)$ respectively. $Q_{\text{num}}^{(t)}(\Lambda|\Lambda')$ is a smoothing function with a zero differential with respect to $\Lambda = \Lambda'$ to ensure that $Q(\Lambda|\Lambda')$ is convex for all Gaussian parameters. $p(\Lambda)$ is prior distribution. For $Q_{\text{num}}^{(t)}(\Lambda|\Lambda')$, statistics of training data are respectively accumulated as follows:

$$\gamma_{lm}^{\text{num}} = \sum_{l=1}^{S} \sum_{t=1}^{T} \gamma_{lm}^{\text{num}}(t),$$

(13)

$$\theta_{lm}^{\text{num}}(\mathcal{X}) = \sum_{l=1}^{S} \sum_{t=1}^{T} \gamma_{lm}^{\text{num}}(t)x_{st},$$

(14)

$$\theta_{lm}^{\text{num}}(\mathcal{X}^2) = \sum_{l=1}^{S} \sum_{t=1}^{T} \gamma_{lm}^{\text{num}}(t)x_{st}^2.$$  

(15)

The numerator statistics are ordinary ML statistics so that the numerator posterior probability of the $t$-th frame in training segment $X_s$ given the Gaussian component $l_m$,

$$\gamma_{lm}^{\text{num}}(t) = \begin{cases} \gamma_{lm}^{\text{num}}(t) & \text{if } l = l_s; \\ 0 & \text{otherwise}. \end{cases}$$

(16)

is nonzero only for Gaussian components in the correct language GMM. The posterior probabilities is computed as

$$\gamma_{lm}^{\text{num}}(t) = \frac{W_l c_{lm} N(x_{st}; \mu_{lm}, \Sigma_{lm})}{\sum_{j=1}^{M} c_{lj} N(x_{st}; \mu_{lj}, \Sigma_{lj})},$$

where $W_l$ is a factor inversely proportional to amount of training data of the $l$-th language [1]. It aims to overcome the problem in the EBW algorithm that the language priors in training data are learned through the accumulation of statistics in numerator and denominator terms.

For $Q_{\text{den}}^{(t)}(\Lambda)$, statistics are the same as (13)–(15) except that the denominator posterior probability is computed as follows:

$$\gamma_{lm}^{\text{den}}(t) = \begin{cases} \gamma_{lm}^{\text{den}}(t)p(X_s|x_{lm})K_{\text{num}}^{\text{num}}(x_{lm}) & \text{if } l \neq l_s; \\ 0 & \text{otherwise}, \end{cases}$$

(16)

which contrarily is zero for Gaussian components in the correct language GMM.

Differentiating (12) with respect to $\Lambda$ and equating to zero, we get update formulæ for mean and variance as follows:

$$\mu_{lm} = \frac{\theta_{lm}^{\text{num}}(\mathcal{X}) - \mu_{lm}^{(t)} + D_{lm}^{(t)} \gamma_{lm}^{\text{num}}(\mathcal{X})}{\gamma_{lm}^{\text{num}} - \gamma_{lm}^{(t)} + \tau + D_{lm}} ,$$

$$\sigma^2_{lm} = \frac{\sigma^2_{lm}(\mathcal{X}) - \sigma^2_{lm}^{(t)} + D_{lm}^{(t)} (\gamma_{lm}^{\text{num}} + \mu_{lm}^{(t)} + r)}{\gamma_{lm}^{\text{num}} - \gamma_{lm}^{(t)} + \tau + D_{lm}} - \mu^2_{lm} ,$$

where $D_{lm}$ is a positive smoothing constant which is set to be greater than 1 twice the smallest value ensuring positive variances, or 2 $E_{lm}$ where $E$ is a positive constant. $\tau$ is an $I$-smoothing constant affecting the narrowness of prior [15].

The inference differs from that of LME [4] in two aspects. First, the numerator statistics in (13)–(15) and the corresponding denominator statistics are accumulated on the whole training set, while they are accumulated only on the support vector set (2) in LME. Second, the margin scale is incorporated into the denominator posterior probability in (16).

4. EXPERIMENTS

The SME method is evaluated on the NIST LRE 2007 corpus [16]. The general language recognition task has 14 target languages. There are 3 test conditions under which nominal durations of test segments are 3, 10 and 30 seconds respectively. Our training data include CallFriend, OHSU 2005, NIST LRE 2007 development data, NIST SRE 2004 and 2006, and OGI 22 languages data. Our development data include NIST LRE 96, 03 and 05 data, and utterances of two languages (Bengali and Thai) in NIST SRE 06 which do not exist in the LRE data. Speech signals are framed at 12.5ms rate and 25ms windows. With cepstral coefficients filtered by RASTA, each frame is converted to a 56-dimensional feature vector composed of 7 static cepstral coefficients and 7-1-3-7 shifted delta cepstral (SDC) coefficients [17]. Non-speech frames are removed with an energy-based voice activity detection (VAD) and feature vectors of each utterance are finally normalized to a standard normal distribution.

We compare the performance of SME with that of the LME and MMI. The three methods optimize GMM parameters based on the whole speech segments. The length of segments can severely affect the recognition performance. Referring to [1], we split each training utterance (after VAD) into a sequence of 3-second segments for training. We first train two gender-dependent GMM-UBMs each with 2048 Gaussian components using the training set. Starting from the GMM-UBMs, GMM parameters (mean and variance) are updated in 10 EM iterations. Top-n training and test strategies are adopted to speed up the processes. We use top-20 Gaussian components per frame in training and top-50 components in test [18]. In EBW, we set $\hat{E} = 2, \tau = 200, K_s = 1/T_s$ where $T_s$ is the number of frames in the speech segment $X_s$. We set $\eta = 1$ and $\epsilon = \infty$ in LME [4], and set $\rho = 1$ in SME. The MMI training process is referred to [1]. The logarithm likelihoods from GMM scoring are calibrated using a back-end which concatenates linear discriminant analysis (LDA) and linear logistic regression (LLR) [19]. The back-end is trained on the development data using FoCal Multi-class toolkit [20].

We first study the margin change with EM iterations in training methods. According to (1) and (9), the margin among the training
set is defined as
\[ M = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{T_s} \left[ \log p(X_s | l_s) - \max_{l \neq l_s} \log p(X_s | l) \right]. \]

Fig 1 compares the margin change with EM iterations in the three methods on the training set. For each method, the margin monotonically increases with the increase of EM iterations, which indicates the model parameters better fit the training set. After each iteration, SME obtains the largest margin and MMI the smallest, which denotes SME optimizes the margin more effectively than the other two methods.

Fig 2 compares EER change with EM iterations in the three methods on the 30-second data. For each method, the EER is reduced with the increase of EM iterations. Performance of the methods can quickly reach convergence in 10 EM iterations. Among them, SME and LME exhibits faster convergent speed than MMI. SME obtains the best EER performance than the other two methods. Table 1 summarizes EERs of the conventional ML training method and the three methods under three test conditions (30s, 10s and 3s). Each of the three methods is more time-consuming than ML, spending around 12 times longer training time. Results show that SME effectively improves the performance from LME. In comparison with MMI, SME obtains relative EER improvement of 22%, 15% and 7% for 30s, 10s and 3-second data respectively.

Table 1. Equal error rates (in %) of ML, MMI, LME and SME.

<table>
<thead>
<tr>
<th>Method</th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>5.53</td>
<td>9.95</td>
<td>20.11</td>
</tr>
<tr>
<td>MMI</td>
<td>5.11</td>
<td>9.45</td>
<td>19.56</td>
</tr>
<tr>
<td>LME</td>
<td>4.28</td>
<td>8.59</td>
<td>18.76</td>
</tr>
<tr>
<td>SME</td>
<td>3.98</td>
<td>8.01</td>
<td>18.23</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

We proposed a soft margin estimation of GMM parameters for spoken language recognition. A loss-scaled margin is defined to incorporate negative samples among training data into the parameter estimation. An unconstrained optimization is derived with a penalty function and the GMM parameters are estimated with EBW. Results show that the SME method effectively optimizes the margin on training data and improves the EER performance on the NIST LRE 07 task. We would like to see our future work to include formulation of convex optimization and comprehensive study of margin definition.

6. REFERENCES