INTERSPEECH 2009 BRIGHTON

Large Margin Estimation of Gaussian Mixture Model Parameters with Extended Baum-Welch for Spoken Language Recognition

Donglai Zhu, Bin Ma, Haizhou Li

Institute for Infocomm Research, Singapore

{dzhu,mabin,hli}@i2r.a-star.edu.sg

Abstract

Discriminative training (DT) methods of acoustic models, such as SVM and MMI-training GMM, have been proved effective in spoken language recognition. In this paper we propose a DT method for GMM using the large margin (LM) estimation. Unlike traditional MMI or MCE methods, the LM estimation attempts to enhance the generalization ability of GMM to deal with new data that exhibits mismatch with training data. We define the multi-class separation margin as a function of GMM likelihoods, and derive update formulae of GMM parameters with the extended Baum-Welch algorithm. Results on the NIST language recognition evaluation (LRE) 2007 task show that the LM estimation achieves better performance and faster convergent speed than the MMI estimation.

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

1. Introduction

In acoustic modeling approaches to spoken language recognition (SLR), acoustic features are modeled by classifiers such as the Gaussian mixture model (GMM) and support vector machine (SVM). The GMM is usually trained with maximum likelihood (ML) estimation. Discriminative training (DT) of GMM has been proved an effective way to improve the performance from the ML training, e.g. maximum mutual information (MMI) estimation [1]. Combination of GMM and SVM also shows the improvement, 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]. The process in DT inspires us to further the study by proposing a new method.

Typical DT methods for GMM, such as MMI and minimum classification error (MCE) [4], usually achieve good performance when the acoustic conditions in test data are well-matched with those in training data because the DT methods are asymptotic upper bounds of the Bayes error given ideal infinite amount of training data [5]. However, the performance often degrades greatly for practical test data that exhibits mismatch with training data. In statistical learning theory [6], it is described as the generalization problem of a learning algorithm. The bound on the test error is a summation of two terms: a training error which is the model's error on training data, and a generalization error which measures the power of the model to deal with test data that is unseen in training data. Conventional training methods, such as MMI and MCE, focus on reducing the training error only, and therefore have weak generalization abilities. The large margin (LM) learning framework differs from them in that it attempts to increase the multi-class separation margin which relates to the bound of generalization error, and one of the most successful examples is the SVM. Recently the LM concept has been used to train the hidden Markov model (HMM) for speech recognition. A large margin estimation and its variant called large relative margin (LRM) estimation were proposed to maximize the minimum margin between HMMs [7][8]. An LM estimation based on the Mahalanobis distance was proposed in [9]. A soft margin (SM) estimation of HMMs was proposed to minimize the empirical loss and maximize the separation margin together [10].

In this paper we propose a large margin estimation of GMM for SLR. The margin of a training sample given a model set is defined as the difference between the likelihood of correct language model and the maximum likelihoods among incorrect language models. The objective function is to maximize the margins over a support vector set that is composed of relatively small positive margin among all segments in training data. The study in [11] reported that MCE does not beat MMI in SLR probably because the generalized probabilistic descend (GPD) algorithm in MCE converges to a local minimum at slow speed in comparison with the extended Baum-Welch (EBW) algorithm in MMI. Therefore, unlike other LM methods that use the GPD algorithm [7][10], we derive update formulae of GMM parameters using the EBW algorithm. Experimental results on the NIST LRE 2007 task show that the proposed LM estimation outperforms the MMI estimation with a faster convergent speed.

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

2. Large Margin GMM

Let's assume there are L target languages to be recognized. The training data is 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 $x_{st} = \{x_{st1}, \ldots, x_{stD}\}$ is a D-dimensional feature vector. Each speech segment is labeled with one of languages denoted as $\mathcal{L} = \{l_s, 1 \leq l_s \leq L, s = 1, \ldots, S\}$. Each language is modeled with a Gaussian mixture model of which parameters are denoted as $\lambda_l = \{c_{lm}, \mu_{lm}, \Sigma_{lm}; m = 1, \cdots, M\}$, where M is the number of Gaussian components, c_{lm}'s are Gaussian mixture weights, $\mu_{lm} = [\mu_{lm1}, \cdots, \mu_{lmD}]^T$ is a D-dimensional mean vector, and $\sigma^2_{lm} = diag\{\sigma^2_{lm1}, \cdots, \sigma^2_{lmD}\}$ is a diagonal covariance matrix.

In statistical learning theory [6], the generalization ability of a classification model is described by its Bayes error. With probability of $1 - \xi$, the upper bound of the test risk (on data

that is drawn i.i.d. as the training data) is given by

$$R(\Lambda) \leq R_{emp}(\Lambda) + \sqrt{\frac{1}{N}\left(h(\log\frac{2N}{h}+1) - \log\frac{\xi}{4}\right)}\ , \quad (1)$$

where $\Lambda = \{\lambda_l, l = 1, \ldots, L\}$ denotes the set of model parameters, $R_{emp}(\Lambda)$ denotes the empirical risk on training data, N is amount of training data, and h is the VC dimension ($h < N$) which equals the maximum number of data samples that can be shattered by models.

The empirical risk can be reduced by fitting the model better with training data using techniques such as increasing model parameters and adopting DT estimation. The second term in Eq. (1) is the generalization risk which measures power of the model to deal with new test data that is unseen in training data. It can be reduced by increasing N or reducing h. When we reduce the empirical risk, the VC dimension h increases because more samples can be shattered by the model that better fits the training data. Accordingly the generalization risk increases. If it causes the total test risk to increase, the over-fitting problem arises which may degrade the recognition performance on new test data that is unseen in training data.

Most conventional training methods attempt to reduce the empirical risk but not the generalization risk in Eq. (1). In the statistical learning theory, the empirical risk is defined as follows [6]:

$$R_{emp}(\Lambda) = \frac{1}{S}\sum_{s=1}^{S}\ell(X_s, \Lambda)\ , \quad (2)$$

where $\ell(X_s, \Lambda)$ is a loss function for the speech segment X_s. The MCE estimation aims to minimize the loss function $\ell(X_s, \Lambda)$ by approximating it with a differentiable function w.r.t. model parameters [4]. In the ML estimation, the loss function is defined as $\ell(X_s, \Lambda) = -\log p(X_s|\lambda_{l_s})$. In the MMI estimation, the loss function is defined as

$$\ell(X_s, \Lambda) = -\log\frac{p(X_s|\lambda_{l_s})p(\lambda_{l_s})}{\sum_{l=1}^{L}p(X_s|\lambda_l)p(\lambda_l)}\ , \quad (3)$$

which is the inverse of the posterior probability of correctly recognizing the training data [12]. MMI training of GMM parameters is presented in details for example in [1]. In practice, several techniques have been used to improve the generalization ability of the methods, e.g. smoothing sigmoid function in MCE [4] and I-smoothing in MMI [12].

In this paper, we attempt to reduce the generalization risk in model training. It is infeasible to minimize the generalization risk directly because of the difficulty of computing the VC dimension. However, it is shown that the VC dimension is bounded by a decreasing function of the separation margin between classes [6], and the concept of large margin has been identified as a unified principle for many different pattern recognition approaches in which one the the most successful examples is the SVM [13]. Several large margin approaches have been studied in the HMM framework. For example, a soft margin was proposed to represent the marge width in [10]. Inspired by the work in [7], we define the margin as a function based on the GMM likelihoods. Let's define the discriminant function of X_s given a language model λ_l as $\mathcal{F}(X_s|\lambda_l) = \log p(X_s|\lambda_l)$. The multi-class separation margin for X_s is defined as

$$d(X_s) = \mathcal{F}(X_s|\lambda_{l_s}) - \max_{1\leq l\leq L, l\neq l_s}\mathcal{F}(X_s|\lambda_l)\ . \quad (4)$$

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

$$\Omega = \{X_s | X_s \in \mathcal{X}, 0 \leq d(X_s) \leq \epsilon\}\ , \quad (5)$$

where $\epsilon > 0$ is a positive number. Each segment in Ω is called a support token which has relatively small positive margin among all segments in the training set $\mathcal{X}$. To achieve better generalization capability, it is desirable to adjust decision boundaries by optimizing the GMM parameters Λ to make all support tokens as far from the decision boundaries as possible. This idea leads to the objective function of the large margin estimation that maximizes the separation margins over the support vector set:

$$\mathcal{O}_{LM}(\Lambda) = \sum_{X_s\in\Omega} d(X_s)\ . \quad (6)$$

3. Parameter Estimation

To derive update formulae of GMM parameters, the objective function Eq. (6) needs to be a continuous and differentiable function. Similar to MCE [4], we approximate the *max* operation in Eq. (4) with a logarithm summation as follows:

$$d(X_s) = \log p(X_s|\lambda_{l_s}) - \frac{1}{\eta}\log\sum_{1\leq l\leq L, l\neq l_s} p(X_s|\lambda_l)^{\eta}\ , \quad (7)$$

where $\eta \geq 1$. When $\eta \to \infty$, the approximation will approach the *max* operation. Segment likelihood $p(X_s|\lambda_l)$ is likely to be underestimated because it is a multiplication of frame likelihoods assuming statistical independence of feature vectors. To overcome this problem, a factor $0 < K_s < 1$ can be applied to $p(X_s|\lambda_l)$ to increase the confusion between numerator and denominator hypothesis [12]. Then Eq. (7) can be rewritten as

$$d(X_s) = \log\frac{a_{(X_s,\Lambda)}}{b_{(X_s,\Lambda)}} = \log\frac{p(X_s|\lambda_{l_s})^{K_s}}{\left(\sum_{\substack{1\leq l'\leq L\\ l'\neq l_s}} p(X_s|\lambda_l)^{K_s\eta}\right)^{1/\eta}}\ . \quad (8)$$

It is desirable to derive the update formulae of GMM parameters with the extended Baum-Welch (EBW) algorithm because Eq. (8) is a rational function within the logarithm operation [14]. Two essential points in the EBW inference are:

1 To maximize $f_{(X,\Lambda)} = \frac{a_{(X,\Lambda)}}{b_{(X,\Lambda)}}$ for positive $a_{(X,\Lambda)}$ and $b_{(X,\Lambda)}$, we can maximize $g_{(X,\Lambda)} = a_{(X,\Lambda)} - kb_{(X,\Lambda)}$ where $k = \frac{a_{(X,\Lambda')}}{b_{(X,\Lambda')}}$ and Λ' is the current value obtained from previous iteration.

2 To ensure that the resulting polynomial is positive, we can add to the expression a constant times a further polynomial which is constrained to be a constant.

The EBW algorithm can be extended to the case of continuous density HMMs by approximating a Gaussian with a discrete distribution [15]. Accordingly, the auxiliary function for the LM estimation in Eqs. (6) and (8) is defined as:

$$Q^{LM}_{(\Lambda,\Lambda')} = Q^{num}_{(\Lambda,\Lambda')} - Q^{den}_{(\Lambda,\Lambda')} + Q^{sm}_{(\Lambda,\Lambda')} + \log p(\Lambda)\ , \quad (9)$$

where $p(\Lambda)$ is prior distribution. Each Q function is similar to that in the ML estimation:

$$Q_{(\Lambda,\Lambda')} = \sum_{m=1}^{M} Q(\gamma_m, \theta_m(\mathcal{X}), \theta_m(\mathcal{X}^2))\ , \quad (10)$$

where

$$\gamma_m = \sum_{X_s \in \Omega} \sum_{t=1}^{T_s} \gamma_{ms}(t) \ , \quad (11)$$

$$\theta_m(\mathcal{X}) = \sum_{X_s \in \Omega} \sum_{t=1}^{T_s} \gamma_{ms}(t) x_{st} \ , \quad (12)$$

$$\theta_m(\mathcal{X}^2) = \sum_{X_s \in \Omega} \sum_{t=1}^{T_s} \gamma_{ms}(t) x_{st}^2 \ . \quad (13)$$

The numerator function $Q^{num}_{(\Lambda,\Lambda')}$ and the denominator function $Q^{den}_{(\Lambda,\Lambda')}$ differ in model networks used to accumulate statistics from the training data. $Q^{sm}_{(\Lambda,\Lambda')}$ is a smoothing function with a zero differential w.r.t. $\Lambda = \Lambda'$ to ensure that $Q^{LM}_{(\Lambda,\Lambda')}$ is convex for all Gaussian parameters, which is defined as follows:

$$Q^{sm}_{(\Lambda,\Lambda')} = \sum_{m=1}^{M} Q(D_m, D_m \mu'_m, D_m(\mu'^2_m + \sigma'^2_m)|\Lambda) \ , \quad (14)$$

where D_m is a positive smoothing constant which is set to be greater than 1) twice the smallest value ensuring positive variances, or 2) $E\gamma^{den}_m$ where E is a positive constant. The prior logarithm likelihood is defined as

$$\log p(\Lambda) = Q(\tau, \tau \frac{\theta^{num}_m(\mathcal{X})}{\gamma^{num}_m}, \tau \frac{\theta^{num}_m(\mathcal{X}^2)}{\gamma^{num}_m}|\Lambda) + k \ , \quad (15)$$

where k is a normalizing term and τ is an I-smoothing constant affecting the narrowness of the prior [12].

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 lm,

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

is nonzero only for Gaussian components in the correct language GMM. The denominator posterior probability is computed as follows:

$$\gamma^{den}_{lms}(t) = \begin{cases} \frac{\gamma_{lms}(t) p(X_s|\lambda_l)^{K_s}}{\left(\sum_{\substack{1 \le l' \le L \\ l' \ne l_s}} p(X_s|\lambda_{l'})^{K_s \eta} \right)^{1/\eta}} & \text{if } l \ne l_s; \\ 0 & \text{otherwise.} \end{cases} \ ,$$

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

The EBW algorithm has a problem that the language priors in training data are learned through the accumulation of statistics in numerator and denominator terms. Therefore we equalize training data of languages by weighting the posterior probabilities $\gamma_{lms}(t)$ with a factor W_l which is inversely proportional to amount of training data of the l-th language [1].

Maximizing the auxiliary function in Eq. (9), we get update formulae for mean and variance as follows:

$$\hat{\mu}_{lm} = \frac{\theta^{num}_{lm(\mathcal{X})} - \theta^{den}_{lm(\mathcal{X})} + D_{lm}\mu'_{lm} + \tau \frac{\theta^{num}_{lm}(\mathcal{X})}{\gamma^{num}_{lm}}}{\gamma^{num}_{lm} - \gamma^{den}_{lm} + D_{lm} + \tau}$$

$$\hat{\sigma}^2_{lm} = \frac{\theta^{num}_{lm(\mathcal{X}^2)} - \theta^{den}_{lm(\mathcal{X}^2)} + D_{lm}(\sigma'^2_{lm} + \mu'^2_{lm}) + \tau \frac{\theta^{num}_{lm(\mathcal{X}^2)}}{\gamma^{num}_{lm}}}{\gamma^{num}_{lm} - \gamma^{den}_{lm} + D_{lm} + \tau} - \hat{\mu}^2_{lm} \ .$$

In case of $\eta = 1$, the update formulae differ from the MMI formulae in the following two points:

- The denominator in MMI objective function is the likelihood given all possible target languages. In the large margin objective function Eq. (4), the denominator is the maximum likelihood among languages excluding the correct language.
- In MMI update formulae, statistics of training data given current model parameters are accumulated over the whole training data set. In large margin formulae, the statistics are accumulated over the support vector set.

4. Experiments

4.1. Experimental Setup

We evaluate our method on the NIST LRE 2007 corpus [16]. The general language recognition task has 14 target languages to be recognized. There are 3 test conditions in which nominal durations of test segments are 3, 10 and 30 seconds respectively. Our training data includes CallFriend, OHSU 2005, NIST LRE 2007 development data, NIST SRE 2004 and 2006, and OGI 22 languages data. Our development data includes 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 LM training with that of the MMI training. Both MMI and LM training optimize GMM parameters based on the whole speech segments. The length of segments can severely affect the recognition performance. If the length is too long, the separation margin between correct and incorrect language models for the segment will be too large to reach a fast convergence of parameter update. If the length is too short, the separation margin will be unreliable. Referring to [1], we split each training utterance (after VAD) into a sequence of 3-second segments, which are used in the MMI and LM training.

Based on the training data, we first train two gender-dependent GMM-UBMs each with 2048 Gaussian components. In both MMI and large margin training, starting from the GMM-UBMs, GMM parameters (mean and variance) are updated in 10 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 $E = 2$, $\tau = 200$, $K_s = 1/T_s$ where T_s is the number of frames in the speech segment X_s. For LM training, we set $\eta = 1$. The MMI training process is referred to [1]. The support vector set for LM training in Eq. (5) is determined by a positive constant ϵ. Instead of setting a heuristic value for it, we extend the support vector set to all positive separation margins. It is feasible because the segments with bigger margins lead to less confusion between numerator and denominator hypothesis and make smaller contribution to statistics in update formulae.

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 Multiclass toolkit [20].

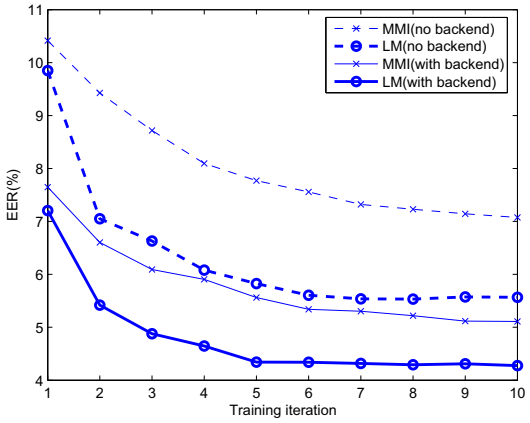

Figure 1: *Comparison of equal-error-rate(EER) change with training iteration in MMI and large margin (LM) training without and with back-end.*

4.2. Experimental Results

We first compare the convergence of MMI and LM training. Fig 1 compares equal error rates (EER) of two training methods after each of 10 update iterations on the NIST LRE 2007 30s data. Two curves are plotted for each method: results without the back-end and results with the back-end. The curves illustrate that, compared with MMI training, the LM training not only effectively improves the performance but also converges the optimization in a fast speed. The MMI training does not optimize the performance until the 10th iteration while the LM training can converge in 5 iterations. Table 1 summarizes EERs of two training methods in three test conditions (30s, 10s and 3s). Both results without back-end and results with back-end show that the LM training achieves better performance than the MMI training in all test conditions. Relative reduction of EER from MMI to LM shows that the LM estimation achieves better improvement on longer test segments.

Table 1: *Equal error rates and relative reduction in % for MMI and large margin (LM) training without and with back-end.*

	30s	10s	3s
MMI (no backend)	7.07	10.72	20.76
LM (no backend)	5.57	9.09	19.59
Relative reduction	21.22	15.21	5.64
MMI (with backend)	5.11	9.45	19.56
LM (with backend)	4.28	8.59	18.76
Relative reduction	16.24	9.10	4.09

5. Conclusions

We proposed a large margin estimation of GMM parameters with the extended Baum-Welch algorithm for spoken language recognition. The training method maximizes the multi-class separation margin to reduce the generalization risk which measures the power of model to handle new test data that is unseen in training data. We formulate the margin as a rational function of GMM likelihoods and optimize the parameters with the EBW algorithm. Results on the NIST LRE 2007 task show that the large margin estimation outperforms the MMI estimation with fast convergence. Future works include: 1) study of ways to handle misrecognition segments in training data, and 2) approximation of the objective function in Eq. (6) with a better differentiable function.

6. References

[1] Matějka P., et al., "Brno university of technology system for NIST 2005 language recognition evaluation", in *IEEE Odyssey: The Speaker and Language Workshop*, 2006.

[2] Castaldo F., et al., "Acoustic langauge identification using fast discriminative training", in *Interspeech*, pp. 346-349, 2007.

[3] Campbell W. M., et al., "A covariance kernel for SVM language recognition", in *ICASSP*, 2008.

[4] Katagiri S., Juang B.-H. and Lee C.-H., "Pattern recognition using a generalized probabilistic descent method", in *Proc. IEEE*, Vol. 86, No. 11, pp. 2345-2373, 1998.

[5] Schluter R. and Ney H., "Model-based MCE bound to the true Bayes' error", in *IEEE Signal Process. Lett.*, Vol. 8, No. 5, pp. 131-133, 2001.

[6] Vapnik V. N., *Statistical Learning Theory*, New York:Wiley, 1998.

[7] Jiang H., Li X. and Liu C., "Large margin hidden Markov models for speech recognition", in *Trans. on Audio, Speeach and Language Proc.*, Vol. 14, No. 5, pp. 1584-1595, 2006

[8] Liu C., Jiang H. and Rigazio L., "Recent improvement on maximum relative margin estimation of HMMs for speech recognition", in *ICASSP*, pp. I-269-272, 2006.

[9] Sha F., *Large margin training of acoustic models for speech recognition*, PhD thesis, University of Pennsylvania, 2007.

[10] Li J., *Soft margin estimation for automatic speech recognition*, PhD thesis, Georgia Institute of Technology, 2008.

[11] Burget L., Matějka P. and Cernocký J., "Discriminative training techniques for acoustic language identification", in *ICASSP*, pp. I-209-212, 2006.

[12] Povey D., *Discriminative training for large vocabulary speech recognition*, PhD thesis, Cambridge University, 2004.

[13] Smola A. J., et al., Eds., *Advances in Large Margin Classifiers*, Cambridge, MA: MIT Press, 2000.

[14] Gopalakrishnan P. S., et al., "A generalization of the Baum Algorithm to rational objective function", in *ICASSP*, pp. 631-634, 1989.

[15] Normandin Y. and Morgera S. D., "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition", in *ICASSP*, pp. 537-540, 1991.

[16] NIST, *The 2007 NIST language recognition evaluation plan*, Available from http://www.itl.nist.gov/iad/mig/tests/lre/2007.

[17] Torres-Carrasquillo P. A., et al., "Approaches to langauge identification using Gaussian mixture models and shifted delta cepstral features", in *ICSLP*, 2002.

[18] Torres-Carrasquillo P. A., et al., "The MITLL NIST LRE 2007 language recognition system", in *Interspeech*, 2008.

[19] Matějka P., et al., "BUT language recognition system for NIST 2007 evaluation", in *Interspeech*, pp. 739-742, 2008.

[20] Brümmer N., et al., "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006", in *IEEE Trans. on Audio, Speeach and Language Proc.*, Vol. 15, No. 7, pp. 2072-2084, 2007.