SAMPLE ITERATIVE LIKELIHOOD MAXIMIZATION FOR SPEAKER VERIFICATION SYSTEMS

Guillermo Garcia, Thomas Eriksson

Communication Systems Group, Department of Signals and Systems,
Chalmers University of Technology, 412 96 Göteborg, Sweden
phone: + (46) 31 772 18 21 fax: +(46) 31 772 17 48
email: em1guill@chalmers.se, thomase@chalmers.se

ABSTRACT

Gaussian Mixture Models (GMMs) have been the dominant technique used for modeling in speaker recognition systems. Traditionally, the GMMs are trained using the Expectation Maximization (EM) algorithm and a large set of training samples. However, the convergence of the EM algorithm to a global maximum is conditioned on proper parameter initialization, a large enough training sample set, and several iterations over this training set. In this work, a Sample Iterative Likelihood Maximization (SILM) algorithm based on a stochastic descent gradient method is proposed. Simulation results showed that our algorithm can attain high log-likelihoods with fewer iterations in comparison to the EM algorithm. A maximum of eight times faster convergence rate can be achieved in comparison with the EM algorithm.

1. INTRODUCTION

Speaker recognition systems are an important part of biometric systems due to the easy deployment and being less invasive compared to other systems. Speaker recognition is the process of automatically recognizing who is speaking based on information provided by speech signals. The main technique is to find a set of features that best represents a specific speaker voice. Speaker recognition systems can be categorized depending on their tasks in speaker identification (SID) and speaker verification (SV) systems. SID systems assign an input utterance from an unknown speaker to one of the predefined known speaker models in the system. Conversely, SV systems are employed to validate whether the speaker is who he or she claims to be. In this work, we will focus on SV systems, although the work can also be extended to SID systems.

Speaker recognition framework can roughly be divided in two phases independent of the task; enrollment and classification. In the enrollment phase, a set of features are extracted from the speech and then used to create the Gaussian Mixture Model (GMM) for each speaker in the database. The Expectation Maximization (EM) algorithm [1] is used to estimate the parameters of the GMMs. In the classification phase, the likelihoods of the speaker models for a sequence of test features from an unknown speaker is computed. Based on the likelihoods obtained, the system makes a decision, i.e., accepted/rejected or identified.

For some time now, the GMMs have been the dominant modeling technique in speaker recognition systems with the EM algorithm as the base for the estimation of the parameters. Other techniques like Gibbs sampling and the methods based on singular value decomposition have been used for the estimation of the parameters [2]. Additionally, discriminative GMM modeling techniques have also been proposed [3, 4], aiming to enhance the specific characteristics of the speaker in the modeling process. Furthermore, other discriminative modeling techniques like support vector machines (SVM) [5] and neural networks [6] are combined with the GMMs as a post-processing stage in order to improve the performance of the system.

While the EM algorithm is a well established modeling technique in speaker recognition, the algorithm requires a large sample set, several iterations and a specific initialization to achieve convergence. The EM algorithm yields a monotonically increasing sequence of likelihoods as the number of iterations increases, implying that the algorithm converges to a stationary point in the likelihood function but does not guarantee that the global maximum will be achieved [7]. Several methods have been proposed to tackle the convergence problem to local maximums. Among these methods, we must emphasize the algorithms that replace or enhance parts of the EM [8] and the ones that accelerate the convergence rate of the EM [9].

This paper aims at developing a competitive algorithm against the EM algorithm in terms of convergence rate and reliability (i.e., high likelihood values at convergence). Simulation results showed that our proposed algorithm can attain higher log-likelihoods with fewer iterations in comparison to the EM algorithm even with random initialization of the parameters.

2. SPEAKER VERIFICATION FRAMEWORK

2.1 Design Phase

In text-independent speaker verification systems, speaker GMMs have become the dominant approach over the last years [1]. Most of the actual verification systems are based on the GMMs or in combination with other classifiers. GMMs can be defined as

$$p(x_t|\lambda) = \sum_{k=1}^{K} w_k \mathcal{N}(x_t|\mu_k, \mathbf{C}_k), \quad (1)$$

i.e., a weighted sum of Gaussian distributions $\mathcal{N}(x_t|\mu_k, \mathbf{C}_k)$, where μ_k is the mean vector, $\mathbf{C}_k$ is the covariance matrix and w_k is the weight of the k-th Gaussian distribution. The GMM can also be defined by a set of parameters, i.e., $\lambda = \{w_k, \mu_k, \mathbf{C}_k\}_{k=1}^{K}$. In this work, we will use GMMs with diagonal covariance matrices

$$\mathbf{C}_k = \mathrm{diag}(\sigma_{k,1}^2, \ldots, \sigma_{k,D}^2), \quad (2)$$

© EURASIP, 2010 ISSN 2076-1465

where $\{\sigma_{k,d}^2\}_{d=1}^D$ is the variance of the k-th Gaussian distribution at the d-th dimension.

In SV systems, two models are defined: the impostor model and the target model. The impostor model also known as the Universal Background Model (UBM) [10] is first trained using the EM algorithm and a pool of speakers different from the speaker we would verify. Then, the speaker model is derived from adapting the mean vectors of the UBM using *Maximum a Posteriori* (MAP) [11] and their own set of features.

2.2 Classification Phase

In the classification phase, the extracted features from a speaker test utterance $\{x_t\}_{t=1}^T$ are compared against the speaker GMM stored in the database. The likelihood of a given speaker model λ for the input utterance is computed as follows

$$\mathcal{L}(x|\lambda) = \sum_{t=1}^{T} \log p(x_t|\lambda). \tag{3}$$

Specifically, SV is a statistical hypothesis test between two hypothesis [12]. The two hypothesis are the target and the impostor model trained in the enrollment phase, and each trial consists of a speaker test utterance and a claimed identity. From each trial, a log-likelihood ratio is computed and a score Θ is determined as

$$\Theta = \log\left(\frac{p(x|\lambda)}{p\left(x|\hat{\lambda}\right)}\right); \qquad \Theta \underset{\text{reject}}{\overset{\text{accept}}{\underset{<}{\geq}}} \tau \tag{4}$$

$$\Theta = \mathcal{L}(x|\lambda) - \mathcal{L}\left(x|\hat{\lambda}\right), \tag{5}$$

where λ denotes the hypothesis to accept an utterance $\{x_t\}_{t=1}^T$ as being produced by the target speaker model. $\hat{\lambda}$ denotes the hypothesis to reject an utterance $\{x_t\}_{t=1}^T$ as being produced by the target speaker and τ is the threshold that minimizes the expected cost of errors. The greater the score obtained, the more likely that the trial is the target speaker.

3. EM ALGORITHM

As mentioned before, the EM algorithm attempts to model the speaker (i.e., determine the underlying pdf of the speaker features) by an iterative maximum likelihood estimation of the parameters of the GMMs. Table 1 describes the EM algorithm consisting of two steps: the Expectation E-step and the Maximization M-step.

The EM algorithm is highly dependent on the initialization procedure; hence the starting positions of the mixture components can determine the convergence rate of the algorithm. A comparison between a random initialization and an initial selection of mixture components (i.e., the mean vectors) close to the speaker pdf shows that the highest likelihood and faster convergence rate was achieved when the mean vectors of the mixtures were initially selected [7]. The initialization algorithm applied in this work consists of K mixture components with weights $\{w_k\}_{k=1}^K = 1/K$.

The mean vectors of each mixture component were initialized using clustering techniques (e.g., vector quantization [13] or K-means [14]) and the covariance matrix $\{\mathbf{C}_k\}_{k=1}^K$ was set to the global covariance of the database. Using the entire training database, the EM iterates until convergence of the likelihood function is attained.

Table 1: EM Algorithm.

```
E-STEP
1. Using the training database $\{x_t\}_{t=1}^T$.
      Compute the total likelihood $LL_t$
      $LL_t = \sum_{k=1}^{K} w_k \mathcal{N}(x_t|\mu_k, \mathbf{C}_k),\ t = 1,\ldots,T.$
2. Normalize the likelihood.
      Compute $\eta_{k,t} = \frac{w_k \mathcal{N}(x_t|\mu_k, \mathbf{C}_k)}{LL_t}$.
      $k = 1,\ldots,K;\ t = 1,\ldots,T.$
3.   Compute the sum of weights, $\hat{w}_k$
      $\hat{w}_k = \sum_{t=1}^{T} \eta_{k,t},\ k = 1,2,\ldots,K.$
4.   Compute the sum of means, $\hat{\mu}_k$
      $\hat{\mu}_k = \sum_{t=1}^{T} \frac{x_t \eta_{k,t}}{\hat{w}_k},\ k = 1,2,\ldots,K.$
5.   Compute the sum of covariances, $\Sigma_k$
      $\Sigma_k = \sum_{t=1}^{T} \frac{x_t^2 \eta_{k,t}}{\hat{w}_k},\ k = 1,2,\ldots,K.$
M-STEP
1. Compute the new parameter values
    for each GMM component.
    $w_k = \frac{\hat{w}_k}{T}.$
    $\mu_k = \hat{\mu}_k.$
    $\mathbf{C}_k = \Sigma_k - \hat{\mu}_k^2.$
```

4. SAMPLE ITERATIVE LIKELIHOOD MAXIMIZATION ALGORITHM

The Sample Iterative Likelihood Maximization algorithm (SILM) is based on the fact that the negative of the log-likelihood function defined in (3) is convex in its parameters. In order to optimize the parameters, we use a stochastic gradient descent method. The principle is to take steps toward the negative of the gradient to achieve a global minimum [15]. An objective function can be represented as

$$H_\theta(x_t) = f(x_t, \theta), \tag{6}$$

where $f(x_t, \theta)$ is a defined function for the input parameter x at time t and is differentiable with respect to the parameter θ. We can define an iterative updating function for the parameter θ as

$$\theta^{t+1} = \theta^t - \rho \nabla H_\theta, \tag{7}$$

where ρ is a step toward the negative of the gradient of the function $\nabla H(\theta_t)$ defined as

$$\nabla H_\theta = \frac{\partial H_\theta(x_t)}{\partial \theta_t}. \tag{8}$$

As mentioned, the SILM algorithm uses the same principle as stochastic gradient descent methods, holding as an objective function the log-likelihood of training feature samples, and as parameters the mean vectors, weights and covariance matrices of the GMM. Moreover, we must denote that the SILM algorithm computes sample-wise operations, contrary to the EM algorithm that performs operations using the whole database. As an example of the algorithm, we use the mean vector of a GMM component as optimization parameter. Substituting (3) in (7) and (8), we attain

$$\mu_k^{t+1} = \mu_k^t - \rho\nabla\mathcal{L}_{\mu_k}\left(x_t|\lambda\right), \tag{9}$$

$$\tag{10}$$

where

$$\nabla\mathcal{L}_{\mu_k}\left(x_t|\lambda\right) = \frac{\partial\mathcal{L}\left(x_t|\lambda\right)}{\partial\mu_k^t}, \tag{11}$$

$$= \frac{w_k\mathcal{N}\left(x_t|u_k,\mathbf{C}_k\right)}{\sum_{k=1}^{K} w_k\mathcal{N}\left(x_t|u_k,\mathbf{C}_k\right)}\left(x_t - u_k\right). \tag{12}$$

Table 2 describes the SILM algorithm. For simplicity, the algorithm was divided in three parts: definition of the step size, computation of the likelihood, and the updating of parameters. In step 1, we define a decreasing step size as a function of the number of iterations. The number of sample iterations (N) is set by the user. The training sample features $\{x_t\}_{t=1}^{T}$ were randomly selected.
Step 2 includes the required operations to compute a single or individual likelihood component and the total likelihood. Finally, step 3 shows the GMM parameters updating as in (7).

The initialization of the mean vector μ_k, covariance matrices $\mathbf{C}_k$, and weights w_k can be done using random initialization of all the parameters or clustering algorithms as the EM algorithm. Moreover, the SILM algorithm as a step descent method requires defining an initial step size (ρ), and step sizes for the parameters: mean (α), covariance matrices (β) and weights (γ). The step sizes for the different parameters of the GMM are directly proportional to the gradient attained. After determining the step sizes, we can iteratively estimate the parameters of the speaker GMM. An extra requirement is to define a threshold (ε), in order to avoid that the weights of the GMMs becoming negative.

The SILM parametrization as a stochastic method aims at avoiding local maxima that may occur with the EM algorithm. Therefore, the algorithm can achieve higher log-likelihood and convergence rate for a given number of iterations.

Note that in the limit when the step size "ρ" of the SILM tends to zero, the SILM and the EM algorithm will fulfill a similar convergence criterion, so the parameters of the GMM will achieve a stable state at convergence for both algorithms. In general, the stable state for the parameters of the SILM is defined as

$$\theta^{t+1} = \theta^t - \rho\nabla H_\theta, \quad t \longrightarrow N \tag{13}$$

$$\theta^{t+1} = \theta^t - \underbrace{\left(1-\frac{t}{N}\right)}_{\approx 0}\nabla H_\theta,$$

$$\theta^{t+1} = \theta^t.$$

Table 2: SILM Algorithm.

```
1. Step Size Definition
    Define a decreasing step size.
    $\rho_0 = \rho\left(1-\frac{t}{N}\right)$.
2.Likelihood Computation
    Using the training database $\{x_t\}_{t=1}^{T}$.
        a. Select one feature vector $\{x_t\}$.
        b. Compute the likelihood
          for each individual mixture $L_k$.
        $L_k = w_k\mathcal{N}\left(x_t|\mu_k,\mathbf{C}_k\right)$,
          and the weighted or total likelihood
        $L_s = \sum_{k=1}^{K} L_k = \sum_{k=1}^{K} w_k\mathcal{N}\left(x_t|\mu_k,\mathbf{C}_k\right)$.
3. Parameter Updating
        a. Compute the step sizes for means,
           covariance matrices and weights.
        $\rho_{\mu_k} = \alpha\cdot\rho\frac{L_k}{L_s}$.
        $\rho_{\Sigma_k} = \beta\cdot\rho\frac{L_k}{L_s}$.
        $\rho_{w_k} = \gamma\cdot\rho\frac{1}{L_s}$.
        b. Update means, covariance matrices,
           and weights.
        $\mathbf{C}_k = (1-\rho_{\Sigma_k})\mathbf{C}_k + \rho_{\Sigma_k}(x_t-\mu_k)^2$
        $\mu_k = (1-\rho_{\mu_k})\mu_k + \rho_{\mu_k}x_t$.
        $w_k = \max\left\{\varepsilon, w_k + \rho_{w_k}\left(L_k - \frac{1}{k}\sum_{k=1}^{K} L_k\right)\right\}$
```

5. EXPERIMENTAL SETUP

The experiments were conducted using the female speakers from the 2005 NIST-SRE "one two-channel (4-wire) conversation" corpus [16]. Each speech file consists of approximately five minutes one-side telephone conversion. After removing silences at the beginning and end, each speech file was segmented into frames of 25 ms length with an overlap of 10 ms. Each frame was pre-emphasized and Hamming windowed. Then, 13th-order MFCCs are obtained and warped [17] with a 3 seconds Gaussian window. Afterwards, deltas, double deltas and delta Log-Energy were computed, yielding to 40 dimension feature space. Finally, a frame removal algorithm based on the most energetic frame was applied to select the features from the silence and noise. Then, we train 64, 128, 256 and 512-mixtures UBMs using 150 speech files from the female speakers in the training set. A common clustering (i.e., K-means) initialization is done for the EM and the SILM algorithm. The initial SILM parameters for the experiments were set as follows: initial step size $\rho_0 = 0.02$, $\alpha = 0.5$, $\beta = 0.05$ and $\gamma = .0001$.

6. EXPERIMENTAL RESULTS

The results are presented in terms of log-likelihood comparisons since both algorithms maximize the likelihood function and experimental evaluation of the performance in speaker recognition systems showed, that similar performances can be achieved in terms of error probability. Moreover, we assume convergence of the algorithm, when the log-likelihood function achieves a stable state.

Figure 1 shows the log-likelihood as a function of the number of exponential operations for 64, 128, 256 and 512-mixture UBM with clustering initialization for both algorithms (initialization operations are not shown). We can observe that with fewer operations or iterations, the SILM achieves convergence faster and attains higher log-likelihood than the EM algorithm. Although the difference in convergence rate between the SILM and the EM algorithm decreases as the number of mixture components increases, the SILM maintains higher likelihood and convergence rate. Table 3 presents a summary of the results obtained by comparing the EM and the SILM algorithms. The first column addresses the number of the mixture components. The second column indicates the log-likelihood values where we considered convergence is achieved. The third and fourth column show the number of exponential operations required to attain the previous log-likelihood values for the EM and the SILM algorithms, respectively. Finally, the fifth column illustrates the convergence speed for the SILM algorithm in comparison to the EM. We can observe that as the number of mixture components increases, the convergence rate of the EM algorithm increases. The reason is that a large number of parameters and a proper initialization can improve the fitting of the model to the features. As mentioned, the initialization of the EM algorithm plays a big roll in the convergence rate and log-likelihood values at convergence. Conversely, the SILM algorithm depends in minor way on the initialization of the parameters.

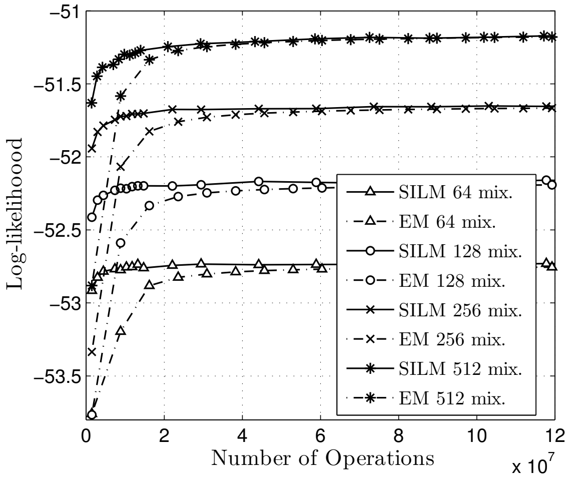

Figure 1: Comparison of the convergence rate and log-likelihood between the EM and the SILM for different number of mixture components with clustering initialization.

Table 3: *Convergence of the log-likelihood as a function of operations when clustering initialization is used.*

No. Mixtures	Log-likelihood	EM Operations	SILM Operations	speed
64	-52.76	63.3×10^6	7.3×10^6	8.6
128	-52.19	59.5×10^6	12.8×10^6	4.64
256	-51.73	54.3×10^6	22.1×10^6	2.45
512	-51.22	41.2×10^6	27.9×10^6	1.47

Figure 2 and 3 present a comparison between the SILM algorithm with random initialization of the parameters and the EM algorithm with clustering initialization for 64 and 512 mixture UBM, respectively. We can observe that even with this type of initialization we are able to achieve faster convergence rate than the EM algorithm and in the case of 64-mixture GMM, a higher log-likelihood is achieved. The shift of the EM log-likelihood values is due to the fact that no log-likelihood is obtained during initialization of the EM algorithm. The initialization operations for the EM are proportional to applying K-means to obtain the mean vector of each mixture component.

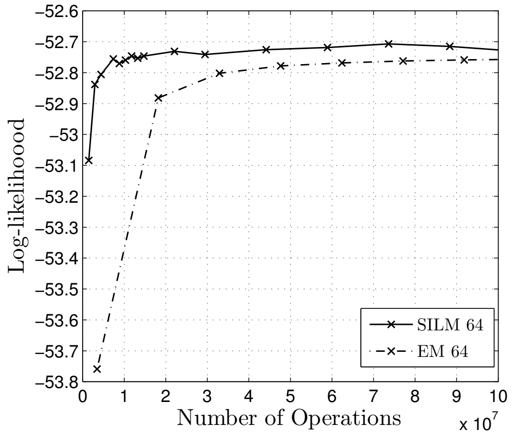

Figure 2: Convergence rate comparison between the EM with clustering initialization and the SILM with random initialization for 64 mixture components.

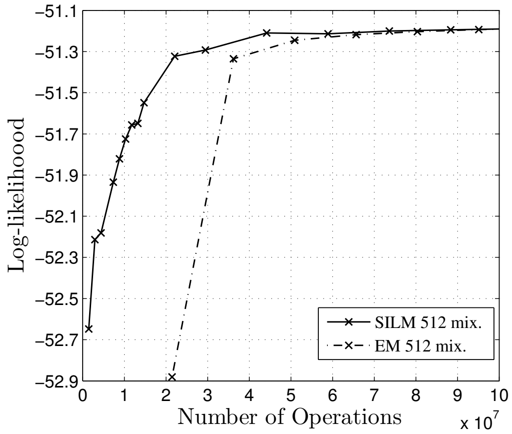

Figure 3: Convergence rate comparison between the EM with clustering initialization and the SILM with random initialization for 512 mixture components.

Although the algorithm presented in this work was used to train the UBM of a SV system, the use of the SILM is not limited to this application. Our intention was to highlight the use of this algorithm with the most time consuming and complex model in speaker recognition. Real time speaker recognition systems and identification systems can also benefit from the use of the SILM algorithm to estimate their parameters.

7. CONCLUSIONS

In this paper, we show an algorithm capable to overcome the convergence limitations of the EM algorithm. The algorithm is based on a step descent method, it achieves faster convergence with fewer number of operations in comparison to the EM algorithm even with random initialization. We also achieve for some cases higher reliability at convergence. This is an ongoing research and requires further study in the sense of using other optimization techniques and increase the robustness of the step size.

REFERENCES

[1] D. Reynolds and R.C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, pp. 72–83, 1995.

[2] Geoffrey Grimmet and David Stirzaker, *Probability and Random Processes*, Oxford University Press, third edition, 2003.

[3] D.E. Sturim, D. Reynolds, et al., "Speaker indexing in large audio databases using anchor models," in *Proc. ICASSP*, 2001, vol. 1, pp. 429–432.

[4] Guillermo Garcia and Thomas Eriksson, "Weight based super-gmm for speaker identification systems," in *Proc. EUSIPCO*, 2008.

[5] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[6] M. W. Mak, W. G. Allen, et al., "Speaker identification using radial basis functions," in *Proc. ICANN*, 1993, pp. 138–142.

[7] McKenzie Patricia and Alder Mike, "Initializing the EM algorithm for use in gaussian mixture modelling," Tech. Rep., University of Western Australia, 1993, 901476.

[8] R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani, "Optimization with em and expectation-conjugate-gradient," in *Proc. ICML*, 2003, vol. 20, pp. 672–679.

[9] Mortaza Jamshidian and Robert I Jennrich, "Acceleration of the em algorithm by using quasi-newton methods," *Journal of the Royal Statistical Society*, vol. 59, no. 3, pp. 569–587, 1997.

[10] Douglas Reynolds, Thomas F. Quatieri, et al., "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, 2000.

[11] Jean-Luc Gauvain and Chin Hui Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[12] Frederic Bimbot, Jean-Francois Bonastre, et al., "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, pp. 430–451, 2004.

[13] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[14] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience Publication, second edition, 2001.

[15] Stephen Boyle, *Convex Optimization*, Cambridge University Press, 2004.

[16] "The NIST 2005 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig//tests/spk/2005/sre-05_evalplan-v6.pdf.

[17] Jason Pelecanos and Sridha Sridharan, "Feature warping for robust speaker verification," in *Proceeding Odyssey*, 2001, pp. 213–218.