Training GMMs for Speaker Verification

Finnian Kelly[†] **and Naomi Harte**[*]

Sigmedia Group
Department of Electronic and Electrical Engineering,
Trinity College Dublin

E-mail: [†]kellyfp@tcd.ie [*]nharte@tcd.ie

Abstract — **An established approach to training Gaussian Mixture Models (GMMs) for speaker verification is via the expectation-maximisation (EM) algorithm. The EM algorithm has been shown to be sensitive to initialisation and prone to converging on local maxima. In exploration of these issues, three different initialisation methods are implemented, along with a split and merge technique to 'pull' the trained GMM out of a local maxima. It is shown that both of these approaches improve the likelihood of a GMM trained on speech data. Results of a verification task on the TIMIT and YOHO databases show that increased model fit does not directly translate into an improved equivalent error (EER) rate. In no case does the split and merge procedure improve the EER rate. TIMIT results show a peak in performance of 4.8% EER at 20 EM iterations and a random GMM initialisation. An EER of 1.41% is achieved on the YOHO database under the same regime. It is concluded that running EM to the optimal point of convergence achieves best speaker verification performance, but that this optimal point is dependent on the data and model parameters.**

Keywords — **Speaker Verification, Gaussian Mixture Model, Expectation Maximisation, Split and Merge**

I Introduction

Gaussian Mixture Models (GMMs) have been used extensively in the field of speaker verification for some time [1, 2]. The standard training procedure uses the Expectation Maximisation (EM) algorithm [3] to fit the GMMs to the speaker training data. Three major problems arise with this approach, as noted in [3, 4, 5]: the sensitivity of the algorithm to initialisation; singularities occurring in the GMM variance; and the tendency of the model to converge to local maxima. The issue of singularities has been successfully addressed by adaptive variance limiting [6]. The other two issues however, remain open problems.

A solution to local maxima convergence by applying a split and merge procedure to the GMM after EM training is presented in [5]. An improvement in speaker verification Equivalent Error Rate (EER) is reported with this technique. In this paper, this technique is implemented and applied to the same test corpus (TIMIT) as [5] and the larger, speaker verification specific, YOHO database.

Both standard EM and split and merge training approaches aim to achieve the same result - a maximum model likelihood given training data. Problems two and three can be seen as interdependent - effective initialisation will prevent the EM converging to local maxima, while a split and merge technique can 'pull' the trained GMM out of a local maxima, thus removing the need for an optimal initialisation strategy.

In the context of speaker verification, the above view presupposes the fact that EER decreases with increased model likelihood. This relationship is not so simplistic however. The results in this paper will show that there is a balance to be struck between improving model fit and maintaining model generality which is dependent on the amount and scope of training data and the model parameters.

In the subsequent section, GMM training via the EM algorithm along with split and merge EM (SMEM) is outlined. EM initialisation strategies are described. In the following section, detailed

experimental results are presented on the TIMIT and YOHO databases and conclusions are drawn from these.

II GMM training for Speaker Models

a) The Gaussian Mixture Model

A speaker's voice can be thought of as occupying some acoustic space, within which lie classes representing different acoustic events (e.g. the utterance of different classes of phoneme). The motivation behind using Gaussian Mixture Models in modelling a speaker's voice is to describe these individual phonetic events via the component Gaussians of the model. The spectral shape of each phonetic class is represented in the GMM via a mean, a component density and a covariance matrix. A GMM is a weighted sum of M component Gaussians given by the equation:

$$p\left(x|\lambda\right) = \sum_{i=1}^{M} w_i g\left(x|\mu_i, \Sigma_i\right) \tag{1}$$

where x is a D-dimensional vector (of speech features in this case). $w_i, i = 1, 2, ..., M$ are the mixture weights of the component Gaussians $g\left(x|\mu_i, \Sigma_i\right), i = 1, 2, ..., M$. Each of these components is a D-variate Gaussian of the form:

$$\frac{1}{\left(2\pi\right)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(x-\mu_i\right)' \Sigma_i^{-1}\left(x-\mu_i\right)\right\} \tag{2}$$

where μ_i is the vector of means and Σ_i is the covariance matrix. The mixture weights w_i must satisfy $\Sigma_{i=1}^{M} w_i = 1$. Thus the complete model is parameterized by the means, weights and covariances, allowing for the compact notation:

$$\lambda = \{w_i, \mu_i, \Sigma_i\} \qquad i = 1, ..., M \tag{3}$$

b) GMM parameter estimation

The goal of training a GMM is to estimate the parameters, λ, which best describe the set of training feature vectors. Given a sequence of T training vectors $X = \{x_1, ..., x_T\}$, the GMM likelihood can be written as:

$$p\left(X|\lambda\right) = \prod_{t=1}^{T} p\left(x_t|\lambda\right) \tag{4}$$

Maximisation of this likelihood function is done using the standard expectation-maximisation (EM) procedure [3]. Given an initial model λ, this iterative algorithm estimates a new model $\overline{\lambda}$ such that $p\left(X|\overline{\lambda}\right) \geq p\left(X|\lambda\right)$. $\overline{\lambda}$ then becomes the initialisation for the second iteration, and so on. Using this algorithm, the model likelihood is guaranteed to increase monotonically. The algorithm is applied via the following update equations:

Mixture Weights:

$$\hat{w_i} = \frac{1}{T}\sum_{t=1}^{T} P\left(i|x_t, \lambda\right) \tag{5}$$

Means:

$$\hat{\mu_i} = \frac{\sum_{t=1}^{T} P\left(i|x_t, \lambda\right) x_t}{\sum_{t=1}^{T} P\left(i|x_t, \lambda\right)} \tag{6}$$

Variances:

$$\hat{\Sigma_i} = \frac{\sum_{t=1}^{T} P\left(i|x_t, \lambda\right) x_t}{\sum_{t=1}^{T} P\left(i|x_t, \lambda\right)} - \hat{\mu_i^2} \tag{7}$$

The *a posteriori* probability is given by

$$P\left(i|x_t, \lambda\right) = \frac{w_i g\left(x_t|\mu_i, \Sigma_i\right)}{\sum_{m=1}^{M} w_m g\left(x_t|\mu_i, \Sigma_i\right)} \tag{8}$$

c) EM training considerations

The EM algorithm guarantees convergence regardless of its initial starting point. However the GMM may have a number of local maxima, any of which the EM may converge to dependent on the initial parameters λ_0, affecting the global likelihood $p\left(X|\lambda\right)$. Thus one approach to optimal EM training is careful selection of λ_0. Three initialisation methods were implemented in this work.

- *Random Initialisation*: Given training data X, M observations are selected at random as the initial component means. The mixture weights are uniform (e.g. $w_i = 1/M, i = 1, ..., M$). The initial covariance matrices for each mixture are equal, with the variance of X across its D dimensions along the diagonal.
- *K-means Initialisation*: A standard K-means algorithm is used to partition the data X into M clusters. The initial component means are then the centres of these clusters, the weights are the normalised number of points within each cluster, and the covariance matrices are the variance of each cluster
- *Iterative Initialisation*: Taking R sets of random initialisation parameters over a number of trials, the means of the random parameter set λ_r which bring about the maximum $p\left(X|\lambda_r\right), r = 1, ..., R$ are passed to a K-means algorithm which clusters the data according to

those means and provides the remaining two initialisation parameters, the weights and covariances.

To demonstrate the outcome of applying these initialisation strategies, a GMM was trained on approximately 20 seconds of speech from a given TIMIT speaker, 25 times, for each of the above schemes. Figure 1 shows the average value of the normalised negative log likelihood (nLogL) in each of the three cases over 21 EM iterations. It is clear that different initialisations will both converge at different rates, and more significantly converge to different local maxima.

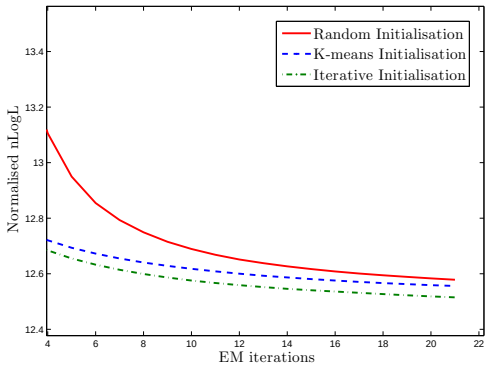

Fig. 1: Convergence behaviour for the EM algorithm with different initialisation methods.

Variance singularities occur in EM training when individual variance entries become very small, which can leading to overfitting and degradation in performance. Typically, GMMs for speaker verification have upwards of 32 mixtures. With training data that may be sparsely distributed, variance singularities become likely, and indeed that was observed in this work. To avoid this problem, adaptive variance flooring [6] was applied according to:

$$if \;\; \sigma_{md}^2 < \gamma S_d^2 \;\; then \;\; \sigma_{md}^2 = \gamma S_d^2 \tag{9}$$

where σ_{md}^2 is the variance of the dth coefficient of the mth mixture and S_d^2 is the overall variance of the dth coefficient. γ is a scaling factor (a value of $5X10^{-3}$) was used here.) This ensures that if the variance of a single element in a given mixture becomes very small relative to the variance of that element across the mixtures, then it is floored to a fixed value.

As discussed, a second approach to the issue of local maxima EM algorithm convergence is to run the algorithm as normal, then to adjust the converged GMM parameters. A technique by Ueda [7] incorporates a split-and-merge procedure into the standard EM algorithm (SMEM). The parameters of the GMM are adjusted by simultaneously merging two mixtures and splitting another such that the number of mixtures M is constant. The motivation being that a GMM will typically model with a redundant number of mixtures in areas of feature space with a high density of data and with too few in areas of low density, widely distributed data. Figures 2 and 3 show how the SMEM operation moves GMM mixtures from high density areas through areas of lower likelihood into low density regions, improving the global likelihood. Zhang [5] introduced modifications to SMEM for speaker GMMs, which were applied in this work.

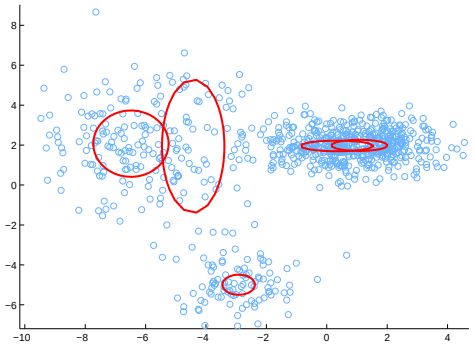

Fig. 2: A 5 mixture GMM trained with the standard EM algorithm on synthetic 2-dimensional data. Normalised nLogL = 4.1

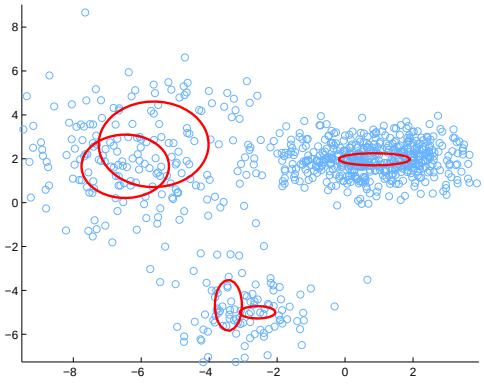

Fig. 3: Trained GMM from Fig 2 after applying a SMEM procedure. Normalised nLogL = 3.8

The split and merge procedure:

- **Merge Criterion**: The goal of the merge step is to combine two mixtures in a densely populated area which are in some way carrying redundant information. To evaluate this, a 'cross-correlation' of posterior probability vectors for each pair of mixtures is computed. Thus the pair of mixtures with the most closely aligned posterior probability vectors will have the highest correlation value.

The merge criterion is given by:

$$J_{merge}(i,j,\lambda) = \frac{P_i(\lambda)^T P_j(\lambda)}{\|P_i(\lambda)\|\|P_j(\lambda)\|} \quad (10)$$

where i and j are mixtures in the model λ and $P_i(\lambda)$ is an M-dimensional vector representing the posterior probabilities of the ith mixture. The denominator term scales the criterion to avoid a bias towards merging larger classes. T denotes the vector transpose and $\|.\|$ the vector norm.

- **Merge Action**: Given two mixtures, i and j, to be merged, the parameters of the merged mixture i' are initialised as linear combinations of i and j as follows:

$$\alpha_{i'} = \alpha_i + \alpha_j \;\; and \;\; \lambda_{i'} = \frac{\alpha_i\lambda_i + \alpha_j\lambda_j}{\alpha_i + \alpha_j} \quad (11)$$

where the mixing proportion of the combination is estimated as the posterior over the data:

$$\alpha_m = \frac{1}{T}\sum_{t=1}^{T} P(m|t_n,\lambda) \quad (12)$$

- **Split Criterion**: The *Kullback Divergence* between two distributions can be used as a measure of their separation. Here the *local Kullback Divergence* is defined as the distance between the local data density around the mth mixture and the density of the mth mixture. It is given by:

$$J_{split}(k,\lambda) = \frac{1}{D_{norm}}\int f_m(x,\lambda)\, log\frac{f_m(x,\lambda)}{p_m(x,\lambda_m)}dx \quad (13)$$

where D_{norm} is simply the normalised number of observations in the mth component and where the local data density is given by:

$$f_m(x,\lambda) = \frac{\sum_{n=1}^{N}\delta(x-x_n)P(m|x_n,\lambda)}{\sum_{n=1}^{N}P(m|x_n,\lambda)} \quad (14)$$

Thus the mixture with the largest $J_{split}(k,\lambda)$ has the worst estimate of the local data density and should be split.

- **Split Action**: Given a mixture, k, to be split, the new models j' and k' are initialised by:

$$\alpha_{j'} = \alpha_{k'} = \frac{\alpha_k}{2} \;\; \lambda_{j'} = \lambda_k + \epsilon \;\; and \;\; \lambda_{k'} = \lambda_k + \epsilon' \quad (15)$$

The complete algorithm can then be described as:

1. Perform the standard EM on initialised parameters λ_0 until convergence. Let λ be the estimated parameters
2. Compute J_{split} and J_{merge} for every pair of mixtures in λ. Sort SMEM candidates according to these criteria. Then let $\{i,j,k\}_c$ denote the cth candidate, where $k_c \notin \{i,j\}_k$.
3. For $c = 1,...,C_{max}$, complete a merge and split according to equations (11) and (15) to estimate a new set of parameters λ. Apply a *partial EM* procedure (full details given in [7]) to force the sum of posterior probabilities of i',j' and k' to be equal the sum of posterior probabilities of i,j and k before the split and merge. This way, the new parameters for models i',j' and k' can be reestimated without affecting the other models. Let $\overline{\lambda}$ be the updated parameters. If $p(X|\overline{\lambda}) > p(X|\lambda)$, then set $\overline{\lambda}$ as the new parameters and go back to Step 2.
4. Stop with the current parameters

III Experiments and Results

a) TIMIT verification task

A SMEM training approach was applied to a speaker verification task by Zhang in [5]. The aim of the initial implementation in this paper was to replicate the results by Zhang. As such, the experiment was designed to follow his method as closely as possible. The choice of model parameters, size of training and test sets used here are consistent with his work. However, information regarding the method of EM initialisation, number of iterations and the exact make up of the training and test sets were not specified in his paper. This is a likely cause of discrepancy in results.

The TIMIT corpus [8] is a speech recognition database, divided into training and test sets. 200 (120 male and 80 female) speakers evenly distributed across dialect regions were taken from the TIMIT training set as subjects. Given the 10 sentences per speaker, 5 (3 SI and 2 SX) sentences per speaker were chosen as training material, with the remainder reserved for testing. After removing silences, standard 12-dimensional MFCCs were extracted from the speech and GMMs with 50 Gaussian mixtures and diagonal covariance matrices were trained for each speaker. The total length of training material amounted to approximately 15 sec per speaker. The models were initialised using the iterative K-means method and EM algorithm was run for 50 iterations. For the SMEM computation, C_{max} was taken as 5.

The average negative log likelihood (nLogL), normalised for length of training vector, for the trained EM and SMEM models is given in Table 1. It is evident that the SMEM procedure provides a more accurate model.

EM iterations	**EM**	**SMEM**
50	12.46	12.43
20	12.55	12.51
5	12.75	12.63
Initialisation		
Iterative K-means	12.55	12.51
K-means	12.59	12.55
Random	12.62	12.58

Table 1: Negative log likelihoods normalised by data length. (Note: the 'EM iterations' results used an iterative K-means initialisation method and the 'Initialisation' results were based on 20 EM iterations)

A universal background model (UBM) [2] of 1024 mixtures was trained with approximately 30 mins of speech evenly distributed among 300 speakers from inside and outside the training set. To avoid biasing, the male-female ratio of 3:2 was preserved.

The testing set for each speaker was their 5 remaining sentences unused in training. For false acceptance testing, each speaker was tested against a randomly chosen test sentence from every other speaker. Thus for each of the 200 speakers there were 199 false acceptance tests. To test the false rejection rate, for each speaker, each of the 5 test sentences were segmented into 5 parts, $n_1, ..., n_5$. From these segments another 5 were composed by 50% overlapping of n_1 and n_2, n_2 and n_3 etc. This scheme lead to a total of 50 false rejection test sentences per speaker. These test set designs were used by Zhang, and so were chosen here in an effort to make our results comparable.

The scoring was done using a basic likelihood ratio framework [2]:

$$\Lambda_s(X) = \log p(X|\lambda_s) - \log p(X|\lambda_{UBM}) \qquad (16)$$

where $p(X|\lambda_s)$ is the likelihood score (4) for speaker s on data X and $p(X|\lambda_{UBM})$ is the likelihood score for the UBM on data X. Then $\Lambda_s(X)$ is the *world normalised* score for speaker s on data X. No further normalisations were applied to the likelihood score.

Table 2 gives the EER rates for the TIMIT testing.

Given these results, and the fact that the SMEM model likelihood is greater (Table 1), it would suggest that the SMEM models have become overtrained. Since the initial EM was run for 50 iterations, the experiment was repeated for EM runs of both 5 and 20 iterations. The outcome is given in Table 2. The performance peaks at 20 EM iterations.

EM iterations	**EM**	**SMEM**
50	5.2%	5.3%
20	5.0%	5.1%
5	5.1%	5.2%
Initialisation		
Iterative K-means	5.0%	5.1%
K-means	4.8%	4.9%
Random	4.8%	4.9%

Table 2: Equivalent Error Rates. (Note: the 'EM iterations' results used an iterative K-means initialisation method and the "Initialisation' results were based on 20 EM iterations)

Zhang quotes an EER of 6.2% for EM and 4.0% with SMEM. While the EM performance here exceeds this, no improvement was seen for SMEM. These results do show however, that the standard EM can match those of SMEM. Furthermore, considering the limited speaker data, deviations in the composition of the training and testing sets could be expected to influence the EER.

Although increasing the EM iterations and applying SMEM increases the model fit, this does not necessarily translate to a reduction of the EER. Drawing from initialisation testing (Figure 1), which showed the relative increase in model fit with an iterative K-means initialisation approach, the experiment was rerun with each of the three EM initialisations; Random, K-means and Iterative K-means with 20 EM iterations. The model fit (normalised nLogL) and EER is given in Table 2. As with the previous findings, the increased initial model fit in fact degrades the verification performance. It would appear that the choice of these training parameters are data dependent. To illustrate the variation in results with test material, the false rejection test was rerun using 5 whole sentences per speaker (rather than 50 segments as previously). The EER fell to 1.65% and 1.7% for EM and SMEM models respectively. This improvement in performance with test sentence length suggested here was also shown by Reynolds in [4].

b) YOHO verification task

The YOHO database [9] is a speaker verification specific database. It contains speech from 138 (108 males and 30 females) speakers, with 96 training sentences and 40 test sentences provided for each speaker. The experiment was rerun using the same GMM and UBM parameters, training the speakers with their entire training set, and the UBM with a set of 600 sentences evenly drawn from different speakers' training material. The EM algorithm

was initialised randomly and run for 20 iterations. Results are shown in Figure 4.

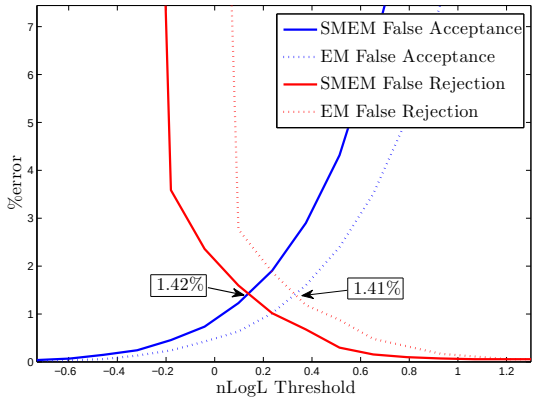

Fig. 4: False Acceptance and False Rejection curves for EM and SMEM models showing the error rate against decision threshold. The EER is the intersection of these curves in each case.

The normalised negative log likelihoods for EM and SMEM models were in this case 12.29 and 12.25 respectively. This again shows that SMEM does not give a better EER despite increasing model fit.

IV Conclusions

The 'issues' of initialisation of GMM parameters for EM training and the EM converging into local maxima are essentially concerned with improving the model fit on the training data. Approaching either of these issues can achieve this goal, i.e. the EM can avoid local maxima with good inital parameters or a technique like SMEM can pull the model out of local maxima without the need for optimal initialisation parameters. In addition, by simply increasing the number of EM iterations, model fit improves.

As discussed however, a better model fit does not guarantee a better EER. A balance has to be struck between training a model that has a sufficient representation of the speaker (keeping the false acceptance rate down) while keeping the model general enough to allow for intra-speaker variance (thus keeping the false rejection rate down)

This relationship between method of EM initialisation, number of iterations (i.e. initial model likelihood) and the resulting EER rate, is a complex one, depending on the number of GMM mixtures and amount and scope of training data etc. This suggests that there is no one training strategy to suit all verification tasks.

The YOHO database results are more representative of large scale results than TIMIT. It would appear from this work that the computational overhead in SMEM is not justified.

In this paper, conventional MFCCs have been used as feature vectors. Future work will investigate the effect of training with conventional features augmented with higher level information. Subsequent work will examine the effect of ageing on the voice, and explore feature sets which provide robustness to this change in a speaker verification context.

This research has been funded by the Irish Research Council for Science, Engineering and Technology.

References

[1] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Commun.*, vol. 17, no. 1-2, pp. 91–108, 1995.

[2] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, M. Chagnolleau, S. Meignier, T. Merlin, O. Garcia, P. Delacretaz, and Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, 2004.

[3] J. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," 1998.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[5] Y. Zhang and M. S. Scordilis, "Optimization of gmm training for speaker verification," pp. 231–236, 2004.

[6] D. James, H.-P. Hutter, and F. Bimbot, "The cave speaker verification project - experiments on the yoho and sesp corpora," in *AVBPA '97: Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication*. London, UK: Springer-Verlag, 1997, pp. 385–394.

[7] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Smem algorithm for mixture models," *Neural Comput.*, vol. 12, no. 9, pp. 2109–2128, 2000.

[8] J. S. Garofolo, "Timit acoustic-phonetic continuous speech corpus," 1993.

[9] J. Campbell and A. Higgins, "Yoho speaker verification," 1994.