Variable Length Mixtures of Inverse Covariances

Vincent Vanhoucke[†⋆], *Ananth Sankar*[⋆]

[†] Department of Electrical Engineering, Stanford University, CA, USA
[⋆] Nuance Communications, Menlo Park, CA, USA
vanhoucke@stanfordalumni.org, sankar@nuance.com

Abstract

The mixture of inverse covariances model is a low-complexity, approximate decomposition of the inverse covariance matrices in a Gaussian mixture model which achieves high modeling accuracy with very good computational efficiency. In this model, the inverse covariances are decomposed into a linear combination of K shared prototype matrices. In this paper, we introduce an extension of this model which uses a variable number of prototypes per Gaussian for improved efficiency. The number of prototypes per Gaussian is optimized using a maximum likelihood criterion. This variable length model is shown to achieve significantly better accuracy at a given complexity level on several speech recognition tasks.

1. Introduction

In a previous paper [1], we introduced the mixture of inverse covariances (MIC) model. It is a very efficient approximation of full covariances in a Gaussian mixture model (GMM). On a variety of speech recognition tasks, we observed a 10% error rate reduction over diagonal covariances at no cost in speed, and as much as 16% error rate reduction at a 50% cost in speed [2].

The evaluation of a Gaussian log-likelihood, using this model, amounts to a scalar product between an extended feature vector and a parameter vector, both of which have dimensionality $D + K$, where D is the input feature dimensionality, and K is the number of prototypes in the MIC model (see Section 2 for a detailed analysis). Given this $D + K$ complexity cost, it is natural to consider optimizing the number of prototypes used on a per-Gaussian basis, so that at a given average complexity level $D + \bar{K}$, Gaussians requiring a more detailed approximation can use a larger number of prototypes than those needing only a coarse approximation.

Solving the variable length problem turns the MIC estimation into a constrained maximum likelihood estimation (MLE), which requires several notable modifications to the algorithm. In Section 2 we review the fixed-length MIC model, sketch its estimation algorithm, and describe the computational complexity associated with it. In Section 3 we describe an extension to variable rate, and detail the constrained MLE procedure for it. In Section 4 we show experimental results that demonstrate the benefits of the model, and conclude in Section 5.

2. Mixtures of Inverse Covariances

A GMM for a D-dimensional input vector $\boldsymbol{o}$, composed of M Gaussians with priors w_i, means $\boldsymbol{\mu}_i$ and covariances Σ_i can be expressed as:

$$f(\boldsymbol{o}) = \sum_{i=1}^{M} w_i \mathcal{N}(\boldsymbol{o}, \boldsymbol{\mu}_i, \Sigma_i) \tag{1}$$

A Mixture of Inverse Covariances is defined by set of K prototype symmetric matrices Ψ_k, such that for each Gaussian i there is a vector $\boldsymbol{\Lambda}_i$ with components $\lambda_{k,i}$ satisfying:

$$\Sigma_i^{-1} = \sum_{k=1}^{K} \lambda_{k,i} \Psi_k \tag{2}$$

2.1. Estimation of the Model

Given the independent parameters $w_i, \boldsymbol{\mu}_i$, and the sample covariance $\bar{\Sigma}_i$, the parameters of the model (Ψ, Λ), with:

$$\begin{aligned} \Psi &= \{\Psi_1, \dots, \Psi_K\} && (3) \\ \Lambda &= \{\boldsymbol{\Lambda}_1, \dots, \boldsymbol{\Lambda}_M\} && (4) \end{aligned}$$

can be estimated jointly using the EM algorithm.The auxiliary function can be written as:

$$Q(\Psi, \Lambda) = \sum_{i=1}^{M} w_i \left[\log |\Sigma_i^{-1}| - \operatorname{Tr}\left(\Sigma_i^{-1} \bar{\Sigma}_i\right)\right] \tag{5}$$

with Σ_i^{-1} as expressed in Equation 2. The joint maximization can be decomposed into two convex optimization problems:

1. maximize $Q(\Psi|\Lambda)$ subject to $\forall i,\ \Sigma_i \succ 0$,
2. maximize $Q(\Lambda|\Psi)$ subject to $\forall i,\ \Sigma_i \succ 0$.

The global maximization problem can be solved by iterating through steps 1 and 2. See [2] for a detailed description of the algorithm.

2.2. Gaussian Evaluation

Using the notations:

$$\begin{aligned} c &= \frac{1}{2}(\log |\Sigma^{-1}| - D \log 2\pi - \boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu}) && (6) \\ \boldsymbol{\omega} &: \omega_k = \frac{1}{2} \boldsymbol{o}^\top \Psi_k \boldsymbol{o} && (7) \\ \boldsymbol{\nu} &= -\Sigma^{-1} \boldsymbol{\mu} && (8) \\ \boldsymbol{o}' &= \begin{bmatrix} \boldsymbol{o} \\ \boldsymbol{\omega} \end{bmatrix} && (9) \\ \boldsymbol{\nu}' &= \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\Lambda} \end{bmatrix} && (10) \end{aligned}$$

The log-likelihood of Gaussian i for observation vector $\boldsymbol{o}$ can be expressed as:

$$\mathcal{L}(\boldsymbol{o}) = c - \boldsymbol{\nu}'^\top \boldsymbol{o}' \tag{11}$$

The cost of evaluating the Gaussian is on the order of $\frac{1}{2} K D^2$ multiplications to compute $\boldsymbol{\omega}$, which is common to all Gaussians (Equation 7), and $K + D$ additional multiplications for each Gaussian being evaluated (Equation 11). In contrast, the cost of evaluating a diagonal Gaussian is $2D$ multiplications per Gaussian.

3. Variable Length Extension

The MIC model constrains the decomposition of the covariances to be of fixed length across the entire GMM. It is possible, however, that some Gaussians would be well estimated with fewer prototypes, in which case computations could be saved from the per-Gaussian scalar product (Equation 11) by truncating the vector. In addition, if the front-end computations are implemented in a "lazy" way, i.e. the actual feature computations are deferred until a Gaussian evaluation actually requires them, additional computations can be saved.

The computational savings of a variable-length model can be even more visible in proportion if the front-end evaluation is inexpensive relative to the per-Gaussian computations. This is especially the case when a subspace-factored MIC model [1] is used, since it reduces the front-end computations by a significant amount.

3.1. Estimation

Denoting by $\boldsymbol{K} = [K_1 \dots K_M]^\top$ a vector listing for $i \in [1, M]$ the length K_i of the MIC decomposition of Gaussian i, the variable-length estimation problem can be expressed as one of constrained optimization:

Maximize $Q(\Psi, \Lambda, \boldsymbol{K})$, subject to:

- a complexity constraint for the average per-Gaussian computational cost: $E[\boldsymbol{K}] = \bar{K}$,
- a complexity constraint for the front-end overhead: $K_i \leq K_{\max}$,
- a feasibility constraint: $K_{\min} \leq K_i$. K_i should be at least 1 for the MIC decomposition to be defined, but $K_{\min} > 1$ might also be used for practical reasons discussed later.

In a manner similar to variable rate vector quantization [4, Chapter 17], the length optimization will be carried out iteratively within the MIC reestimation framework:

1. maximize $Q(\Psi|\Lambda, \boldsymbol{K})$ subject to $\forall i \Sigma_i \succ 0$,
2. maximize $Q(\Lambda|\Psi, \boldsymbol{K})$ subject to $\forall i \Sigma_i \succ 0$,
3. maximize $Q(\boldsymbol{K}|\Psi, \Lambda)$ subject to $E[\boldsymbol{K}] = \bar{K}$ and $K_{\min} \leq K_i \leq K_{\max}$.

Steps 1,2 and 3 are iterated until the Q function reaches its maximum. The first two steps are not different from the fixed-length case. The last one is more difficult: once the optimal Ψ and Λ have been found for a given set of lengths $\boldsymbol{K}^\star$, then we only know $Q(\Psi, \Lambda, \boldsymbol{K}^\star)$. From this data point, deducing the rest of the function $Q(\Psi, \Lambda, \boldsymbol{K})$ for an arbitrary $\boldsymbol{K}$, in order to optimize it, requires finding an optimal set of weights for every $\boldsymbol{K}$. This is prohibitively expensive, since we need to re-run a descent algorithm akin to step 2 for each Gaussian and each value of K_i. Even a search strategy around $\boldsymbol{K}^\star$ would be complex, since there is no guarantee that the function Q_i for a given Gaussian i is convex in K_i.

We can, however, *model* Q given the information we know about it. Section 3.2 describes a parametric model used to represent Q for the purposes of this optimization. The model used is convex, which turns the maximization into a constrained convex optimization problem, which is solved in Section 3.3.

3.2. Parametric Model of Q

Let's assume that an initial length vector $K^\star$ is known, and that steps 1 and 2 were run to estimate:

$$\Sigma^{-1}_{i,K^\star} = \sum_{k=1}^{K^\star} \lambda_{k,i} \Psi_k \qquad (12)$$

We know several things about $Q = \sum_i w_i Q_i(\Psi, \boldsymbol{\Lambda}_i, K_i)$:

- Since the likelihood can only be improved by adding components, Q_i is increasing with K_i,
- $Q_{i,K^\star} = Q_i(\Psi, \boldsymbol{\Lambda}_i) = \log|\Sigma^{-1}_{i,K^\star}| - D$ is known for the current length,
- $Q_{i,1} = Q(\Psi_0, \Lambda_i, 1)$ can be found analytically:

$$Q_{i,1} = \log|\frac{D}{\mathrm{Tr}(\Psi_0 \bar{\Sigma}_i)} \Psi_0| - D \qquad (13)$$

- In the limit, when the number of weights reaches the number of free parameters in the covariance matrix, at $K_{\mathrm{lim}} = \frac{D(D+1)}{2}$, the ML estimate of the covariance is reached exactly:

$$Q_{i,\mathrm{lim}} = Q_i(\Psi, \Lambda_i, K_{\mathrm{lim}}) = \log|\bar{\Sigma}_i^{-1}| - D \qquad (14)$$

From this information, we can build a parametric model of Q_i for all length $K \in [1, K_{\mathrm{lim}}]$. Figure 1 shows how Q behaves on average across all Gaussians in a test GMM used in acoustic modeling. This suggests that a reasonable model for the likelihood would be linearly connecting $\log(Q_K - Q_1)$ with $\log\log K$. For this reason, in the following we used the parametric model:

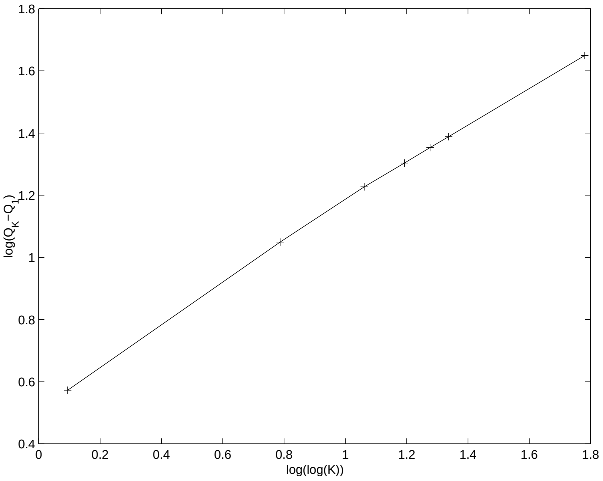

Figure 1: *Plot of* $\log(Q_K - Q_1)$ *against* $\log\log K$. *The approximately affine relationship suggests a simple parametric model for the Gaussian likelihood as a function of* K.

$$Q_i(K) = Q_{i,1} + \alpha_i [\log K]^{\beta_i} \qquad (15)$$

The two free parameters α_i and β_i can be computed for each Gaussian i using a regression on the known values of Q_i: $Q_{i,K^\star}$ and $Q_{i,\mathrm{lim}}$.

3.3. Convex Optimization

The MLE process can now be formulated as:

- maximize: $Q(\boldsymbol{K}) = \sum_i w_i \alpha_i (\log K_i)^{\beta_i}$,
- subject to: $\sum_i w_i K_i = \bar{K}$ and $K_{\min} \leq K_i \leq K_{\max}$.

We will use standard convex optimization methods to solve the problem. First, let's assume that the constraints are not present. In that situation, a standard Newton algorithm can be used to optimize Q [3, Chapter 9]. For that, we compute the gradient $\boldsymbol{\nabla}$ with respect to $\boldsymbol{K}$ and the Hessian H. Note that the Hessian is diagonal here. For simplicity we'll denote by $\boldsymbol{R}$ the diagonal of the inverse of the Hessian. Denoting by $\star$ the Kronecker product of two vectors, the Newton update would be written:

$$\boldsymbol{\Delta}_{\boldsymbol{K}} = -\boldsymbol{R} \star \boldsymbol{\nabla} \qquad (16)$$

The equality constraint $\sum_i w_i K_i = \bar{K}$ is linear. Denoting by $\boldsymbol{\Pi}$ the vector of priors, the constraint can be written as:

$$\boldsymbol{\Pi}^\top \boldsymbol{K} = \bar{K} \qquad (17)$$

The Newton update can be modified simply to incorporate it as follows [3, Chapter 10].

Noting $\boldsymbol{U} = \boldsymbol{R} \star \boldsymbol{\Pi}$:

$$\boldsymbol{\Delta}_{\boldsymbol{K}} = \frac{\boldsymbol{U}^\top \boldsymbol{\nabla}}{\boldsymbol{U}^\top \boldsymbol{\Pi}} \boldsymbol{U} - \boldsymbol{R} \star \boldsymbol{\nabla} \qquad (18)$$

This modification still bears the same convergence properties as the unconstrained update, but preserves the equality constraint by forcing the update to happen in the hyperplane orthogonal to $\boldsymbol{\Pi}$. Indeed we have:

$$\boldsymbol{\Pi}^\top \boldsymbol{\Delta}_{\boldsymbol{K}} = 0 \qquad (19)$$

And thus if $\boldsymbol{\Pi}^\top \boldsymbol{K} = \bar{K}$, then $\boldsymbol{\Pi}^\top (\boldsymbol{K} + \boldsymbol{\Delta}_{\boldsymbol{K}}) = \bar{K}$.

In order to enforce the inequality constraints, we use a barrier method [3, Chapter 11]. The idea is to augment the function to optimize with a family of *barrier functions* which satisfy the inequality constraints by design. The family $\phi(\boldsymbol{K})/t$ parameterized by a parameter t is such that when $t \to +\infty$, the function goes to 0 everywhere in the admissible space, and to $-\infty$ outside of it. Instead of optimizing $Q(\boldsymbol{K})$ directly, t is fixed to some finite value, and $Q(\boldsymbol{K}) + \phi(\boldsymbol{K})/t$ is optimized by only taking the equality constraints into account. t is then increased and the optimization iterated until convergence. This turns the overall problem into a succession of problems which only involve equality constraints, and which we know how to solve.

Here we use the simple log barrier function to ensure $K_{\min} \leq K_i \leq K_{\max}$:

$$\phi(\boldsymbol{K}) = \log(\boldsymbol{K} - K_{\min}) + \log(K_{\max} - \boldsymbol{K}) \qquad (20)$$

4. Experiments

4.1. Experimental Setup

The recognition engine used is a context-dependent hidden Markov model (HMM) system with 3358 triphones and tied-mixtures based on Genones [5]: each state cluster shares a common set of Gaussians called Genone, while the mixture weights are state-dependent. The system has 1500 Genones and 32 Gaussians per Genone.

The test-set is a collection of 10397 utterances of Italian telephone speech spanning several tasks, including digits, letters, proper names and command lists, with fixed task-dependent grammars for each test-set. The features are 9-dimensional MFCC with Δ and Δ^2.

The training data comprises 89000 utterances. Each model is trained using fixed HMM alignments for fair comparison. The Genones are initially trained with full covariances using Gaussian splitting [6]. After the required number of Gaussians per Genone is reached using splitting, the sufficient statistics are collected. In order to train the MIC models, all the Genones are grouped into one large GMM, with Gaussian weights computed from the accumulated prior of all the HMM states corresponding to each Genone. The MIC model is trained in one iteration on this GMM.

The accuracy is evaluated using a sentence understanding error rate, which measures the proportion of utterances in the test-set that were interpreted incorrectly.

4.2. Length Allocation

The length allocation algorithm runs after each iteration of the weight reestimation. Figure 2 shows the likelihood increase during a given run of the length optimization. The first sharp rise in likelihood happens as the Newton algorithm is run for a fixed barrier factor t and corresponds to the initial optimization starting from a uniform length distribution. The second likelihood increase corresponds to the barrier factor being slowly increased, bringing the constrained length distribution closer to its global optimum.

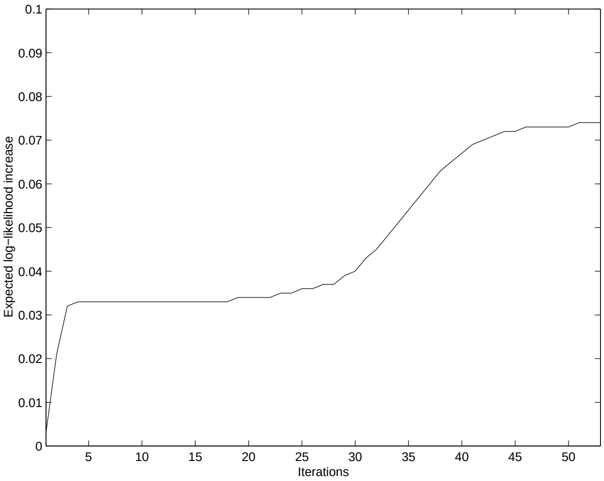

Figure 2: *Likelihood increase as the length allocation algorithm is iterated.*

Figure 3 shows how the allocation algorithm distributes the weights to the various covariances in the GMM in the acoustic model used in our experiments.

Since fewer than 27 weights are used on average, the total number of prototypes that need to be evaluated at each input frame of speech might be less than 27 as well. Thus if the front-end computation is implemented in a lazy way, substantial computational savings can be obtained in addition to the reduction in per-Gaussian computations.

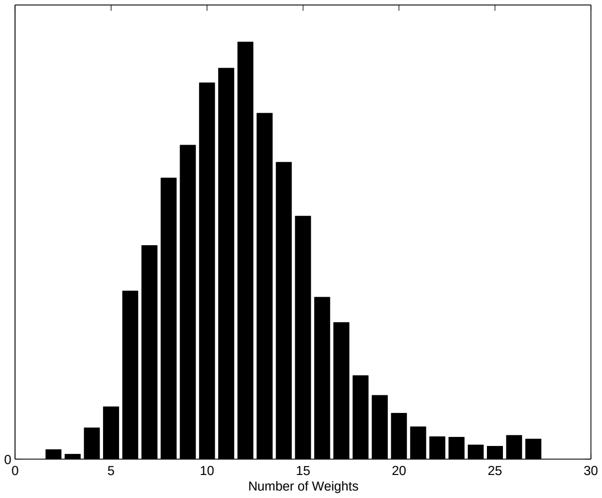

Figure 3: *Histogram of the number of weights allocated per Gaussian by the MLE algorithm. Here, the average number of weights is set to 12, the minimum 2 and the maximum 27.*

4.3. Accuracy

Table 1 shows the error rate achieved on the set of Italian tasks using several setups: the baseline model is a fixed-length model with 12 weights, which is compared with a variable-length model with the same average number of weights. In these experiments, only the Genones corresponding to triphone models were trained using a variable-length optimization. The other Genones in the system were trained with a fixed number of weights equal to the average number of weights. The variable-length model achieves an improved accuracy of about 5.6%. A fixed length model with the same total number of prototypes (18) achieves a 6.3% relative improvement on the same task.

Table 1: *Error rate on a set of Italian tasks.*

Type	$K_{\min}$	K	$K_{\max}$	Error Rate
Diagonal cov.				9.64%
Fixed length		12		9.25%
Variable length	2	12	15	9.04%
Variable length	2	12	18	8.73%
Fixed length		18		8.67%

This demonstrates that a better accuracy can be achieved with the same overall number of Gaussian-dependent parameters.

4.4. Speed / Accuracy Trade-off

Figure 4 shows the speed / accuracy trade-offs attained by the variable-length models. Each curve displays the error rate against the speed of a given system when the level of pruning in the acoustic search is varied. The variable-length system with $K_{\max} = 15$ matches the speed of the 12 weight, fixed-length model at aggressive levels of pruning, while leading to better accuracy for larger pruning thresholds. The variable-length system with $K_{\max} = 18$ matches closely the accuracy of the 18 weight, fixed-length model at large pruning thresholds, while being faster at a given error rate at lower pruning levels. Overall, the variable length systems are capable of achieving trade-offs that were not attained by the fixed-length models.

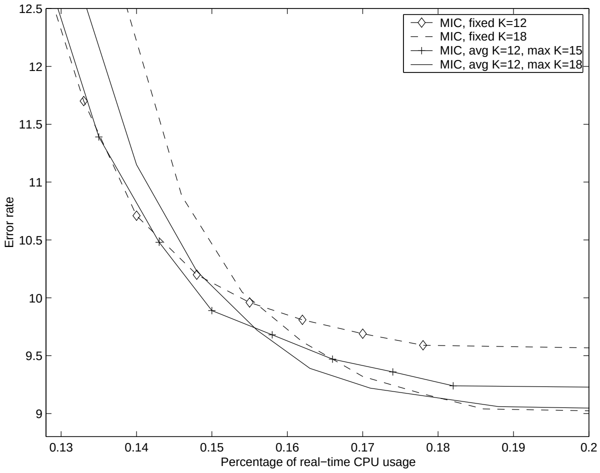

Figure 4: *Speed / accuracy trade-off on the set of Italian tasks. The curves are generated by varying the level of pruning in the acoustic search.*

5. Conclusion

This paper demonstrates that the MIC model can be improved by optimizing the degree of precision by which covariances are approximated on a per-covariance basis instead of globally. An efficient constrained MLE algorithm was proposed to perform this per-covariance weight allocation. Results on a speech recognition task show that the error rate is reduced significantly, and that better speed / accuracy trade-offs can be obtained for a fixed average number of Gaussian-dependent parameters.

6. References

[1] V. Vanhoucke and A. Sankar, “Mixtures of inverse covariances,” in *Proceedings of ICASSP’03*, 2003.

[2] V. Vanhoucke and A. Sankar, “Mixtures of inverse covariances,” submitted to the *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2003.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, draft preprint available on the web at:
http://www.stanford.edu/˜boyd/cvxbook.html, 2003.

[4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[5] V. Digalakis, P. Monaco, and H. Murveit, “Genones: Generalized mixture tying in continuous hidden Markov model-based speech recognizers,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 4, pp. 281–289, 1996.

[6] A. Sankar, “Robust HMM estimation with Gaussian merging-splitting and tied-transform HMMs,” in *Proceedings of ICSLP98*, 1998.