Comparison of Subspace Methods for Gaussian Mixture Models in Speech Recognition

Matti Varjokallio, Mikko Kurimo

Adaptive Informatics Research Centre, Helsinki University of Technology, Finland
Matti.Varjokallio@tkk.fi, Mikko.Kurimo@tkk.fi

Abstract

Speech recognizers typically use high-dimensional feature vectors to capture the essential cues for speech recognition purposes. The acoustics are then commonly modeled with a Hidden Markov Model with Gaussian Mixture Models as observation probability density functions. Using unrestricted Gaussian parameters might lead to intolerable model costs both evaluation- and storagewise, which limits their practical use only to some high-end systems. The classical approach to tackle with these problems is to assume independent features and constrain the covariance matrices to being diagonal. This can be thought as constraining the second order parameters to lie in a fixed subspace consisting of rank-1 terms. In this paper we discuss the differences between recently proposed subspace methods for GMMs with emphasis placed on the applicability of the models to a practical LVCSR system.

Index Terms: speech recognition, acoustic modeling, Gaussian mixture, multivariate normal distribution, subspace method

1. Introduction

Acoustic modeling of an automatic speech recognizer (ASR) is typically done via continuous-density HMMs with Gaussian mixture models (GMM) as observation probability density functions for tied HMM states. Using unrestricted covariance matrices in this framework is possible, but for state-of-the-art large-vocabulary continuous speech recognizer (LVCSR) with many GMMs, the vast amount of parameters may lead to estimation problems. It is also at the moment hard to achieve real-time performance using full covariances. Instead a typical approach is to use diagonal covariance matrices coupled with a maximum likelihood linear transform (MLLT) [5]. This is a reasonable approximation, because the acoustic feature vectors are typically computed in such a way that the elements are only weakly correlated. This is, however, a global property and typically on a state-level there may be significant correlations between the elements. From the mathematical standpoint, using diagonal covariances equals to constraining the second order parameters to lie in a fixed d-dimensional subspace consisting of rank-1 terms, when considered as matrices. Thus, there might be use for more general subspace methods for the mixtures, that still preserve the possibility to explicitly model correlations between the feature components. Just using any technique for dimensionality reduction isn't reasonable, because the possible real-time requirements have to be taken into account. Care should thus be taken that not only the number of parameters decreases, but also a decrease in the computational cost is guaranteed, while keeping the recognition accuracy reasonable. In recent years there has been much interest towards more general and effective subspace methods for GMMs: EMLLT [8], MIC [14], PCGMM [3], SPAM [3], SCGMM [3]. The differences between these models are discussed and emphasis is placed on the applicability of the models to a practical LVCSR system. We present the first comparison between PCGMM and SCGMM along with a baseline diagonal model on LVCSR tasks for Finnish and English.

1.1. Gaussians as an exponential family

The Gaussian distribution belongs to the family of exponential distributions and the multivariate Gaussian can be written equivalently as an exponential family as:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}^T)} e^{\boldsymbol{\theta}^T \boldsymbol{f}(\mathbf{x})}, \tag{1}$$

where $Z(\boldsymbol{\theta}) = \int_{\mathbb{R}^d} e^{\boldsymbol{\theta}^T \boldsymbol{f}(\mathbf{x})}$ is the normalizer, which guarantees that the result is a valid probability distribution function. The features $\boldsymbol{f}(\mathbf{x})$ and parameters $\boldsymbol{\theta}$ are written as:

$$\boldsymbol{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} \\ -\frac{1}{2}\text{vec}\left(\mathbf{x}\mathbf{x}^T\right) \end{pmatrix} \text{ and } \boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\psi} \\ \text{vec}\left(\mathbf{P}\right) \end{pmatrix}, \tag{2}$$

where the precision matrix $\mathbf{P} = \boldsymbol{\Sigma}^{-1}$ and linear parameters $\boldsymbol{\psi} = \mathbf{P}\boldsymbol{\mu}$ are called either canonical or exponential parameters of the distribution and vec-operator maps either triangular of a symmetric matrix to a vector with the off-diagonal elements multiplied by $\sqrt{2}$. $\mathbf{x}$ is the original feature vector. As discussed in [7], this reparametrization is in many ways more natural than the typical presentation through the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. What is specifically interesting when discussing possible subspace methods for GMMs is that from this formulation we can directly see which parameters appear linearly in the data-dependent terms. For this reason constraining the exponential model parameters to a subspace leads to a decrease not only in the amount of parameters but also to a decrease in the computational complexity of the model. This property makes models that constrain exponential parameters to a subspace appealing for speech recognition purposes.

2. Subspace methods for GMMs

The idea is to tie some or all of the exponential model parameters to a subspace shared by all the Gaussians in the system:

$$\boldsymbol{\theta}_g = \sum_{k=1}^{D} \lambda_g^k \mathbf{b}_k, \tag{3}$$

where $\boldsymbol{\theta}_g$ is some part of the exponential parameter vector for Gaussian g, D the subspace dimensionality, $\mathbf{b_k}$ the basis dimension k and λ_g^k the component parameter for Gaussian g and basis dimension k.

The maximum likelihood training of Gaussian mixtures is typically done by the Expectation-Maximization algorithm. The E-step consists of collecting the sufficient statistics for the given mixture and data; this is straightforward for any model. In the M-step, the model parameters are set to maximize the expected likelihood of the data. This has a closed form solution for diagonal and full covariance models. For subspace constrained models, the parameter optimization becomes harder because the parameter space is in two parts and the basis components don't necessarily correspond to positive definite precisions. The optimization can be performed in a round-robin style by alternating the optimization of the component-wise and global parameters:

$$\text{Maximize } Q\left(\mathbf{\Lambda}|\mathbf{B}\right) \ | \text{Maximize } Q\left(\mathbf{B}|\mathbf{\Lambda}\right), \qquad (4)$$

where Q is the expected likelihood of the data given the model from the last step. Both steps are concave fixing the parameters that aren't to be optimized. The concavity property of the steps isn't quite straightforward and is discussed in [7].

Five methods have been suggested that tie the exponential model parameters to a subspace. They are in the order of growing generality:

- Extended Maximum Likelihood Linear Transform (EMLLT) [8]: The precision matrix is modeled as a sum of rank-1 matrices. This is an extension of MLLT where more directions are added in which to compute the variance.
- Mixtures of Inverse Covariances (MIC) [14]: The precision matrix is modeled as a sum of symmetric matrices. The explained implementation however initializes the basis matrices to be positive definite.
- Precision Constrained Gaussian Mixture Models (PCGMM) [3]: The precision matrix is modeled as a sum of symmetric matrices and the positive definiteness is ensured through valid component parameters.
- Subspace Precision and Mean (SPAM) [3]: The precisions and linear parameters are modeled in subspaces independent of each other. The term SPAM is commonly used when referred to the PCGMM model, but for clarity we distinct these cases here.
- Subspace Constrained Gaussian Mixture Models (SCGMM) [3]: All exponential parameters are modeled in the same subspace.

There are some issues to consider when selecting which type of subspace constraint to use for modeling. As we have to resort to an optimization algorithm in the parameter training, it would be nice if the training could be performed in as few iterations as possible. If the initialization is done wisely, also the basis optimization can be left out.

EMLLT is likely not the best use of per-Gaussian parameters and PCGMM has been found to outperform the EMLLT model in [4]. PCGMM seems also to be slightly better justified than the MIC model; the positive definiteness for the precisions can be ensured through valid coefficients, so the basis doesn't need to be positive definite. It is hard to come up with a good initialization scheme that results in positive definite basis matrices and as stated in [14], the basis has to be trained when using the suggested Kullback-Leibler -based clustering scheme. The PCGMM model can also be initialized using PCA and this allows to leave out the basis optimization [4].

Constraining also the linear parameters to a subspace of their own (SPAM) has been shown to lead to better parameter usage [3] in a restricted grammar task. Constraining the first order parameters to a subspace of their own in a LVCSR context might not be a good idea because the linear parameters aren't very redundant and the amount of states is large, leading easily to decreased state discrimination. As the training needs also a PCGMM model for initialization, we decided to leave the SPAM model out of our considerations.

In [3] the most general SCGMM was found to outperform SPAM and PCGMM models but in a restricted grammar task and the initialization was done by first training PCGMM, then SPAM and using this as an initialization for the SCGMM model. In [9] an easier initialization using PCA with modified norm for the exponential parameters of a full covariance model was used. Results were again given on a restricted grammar task and the models didn't see any training data. In that setup the full covariance model gave the best performance, but that is not always the case with limited training data so it would be interesting to know how well these results generalize to a practical LVCSR system. SCGMM is in some way very interesting because all the parameters are tied.

Following these deductions, the PCGMM and SCGMM models seemed like the most interesting models to try in our speech recognizer. In the following subsections, the models are introduced and the most valid pointers to the literature are given.

2.1. Precision Constrained GMM

The PCGMM model constrains the precision matrices to lie in a shared subspace:

$$\mathbf{P}_g = \sum_{k=1}^{D} \lambda_g^k \mathbf{S}_k, \qquad (5)$$

where the subspace dimensionality D is free to vary between 1 and $d(d+1)/2$ and $\{\mathbf{S}_k\}$ are the symmetric basis matrices. Typically the first element is taken to be positive definite and the subspace to be affine, fixing $\lambda_g^1 = 1$ for all g. The number of parameters becomes $D \times (d(d+1)/2) + G \times (D+d+1)$, where the per-Gaussian cost is linear.

For the initialization, second order statistics are needed and they can be collected using any seed model. Doing a quadratic approximation to the auxiliary function Q one arrives at doing Principal component analysis for the precisions under a 'modified Frobenius' norm. This initialization procedure is explained in [4] and [3] and slightly differently in [10]. It is also possible to do PCA directly on the precisions.

The parameter training has been explained in detail in [3]. An optimization library is typically needed, although in [10] a claimedly faster stand-alone implementation was explained. The basis training typically doesn't have a significant effect on the performance, because all precisions have to stay positive definite when optimizing the basis. We performed only optimization of the component parameters.

2.2. Subspace Constrained GMM

In SCGMM all the exponential model parameters are constrained to lie in a shared subspace:

$$\boldsymbol{\theta}_g = \sum_{k=1}^{D} \lambda_g^k \mathbf{b}_k, \qquad (6)$$

where the subspace dimensionality D is free to vary between 1 and $d(d+3)/2$ and $\{\mathbf{b}_k\}$ are the basis vectors. Typically the first element is selected so that the precision part is positive definite when mapped to a matrix and the subspace to be affine, fixing $\lambda_g^1 = 1$ for all g. The number of parameters becomes $D \times (d(d+3)/2) + G \times (D+1)$ and we note that the per-Gaussian cost is independent of the feature dimensionality because all the parameters are tied.

This model can be initialized through PCGMM and SPAM models as in [3]. The other choice is to do a similar approximation as in the case of PCGMM model. For SCGMM initialization, statistics are naturally needed for both first and second order terms which equals a trained full covariance model. This scheme is explained in [9]. If the data is normalized to have zero mean and unitary variance this corresponds to doing PCA directly on the exponential model parameters.

The parameter training has been explained in detail in [3] and is similar to the PCGMM parameter training. The basis training shouldn't have a significant effect on the performance [9] and so we left it out.

3. Results

3.1. Setup

We provide results for a Finnish and an English LVCSR task. The language-independent issues are explained here and language-dependent issues in the corresponding subsections. The acoustic modeling is based on context-dependent cross-word triphones tied using a decision tree [6]. The features were standard 39-dimensional MFCC with MLLT for all models. A state-based segmentation was obtained using a previous best-performing diagonal model. This segmentation was kept fixed throughout the tests because it has only little effect on the results. The training for all models was done in Viterbi-style. First a diagonal model was trained until convergence. The full covariance model was trained by doing two more EM-iterations using the diagonal models as a seed. The basis for the subspace constrained models were both initialized from this full covariance model using PCA for the parameters as explained. The initial component-wise parameters were trained also from this full covariance model. Then two more EM-iterations were done and the component parameters were updated correspondingly. The basis was kept in its initial form. The variance terms were floored to 0.1 in each iteration and after that the eigenvalues of the sample covariance matrices to 0.05 to avoid singularity issues with full covariance models. The parameter optimization for subspace constrained models was done using limited-memory BFGS algorithm as implemented in the freely available *Hilbert Class Library* [1] -package. The latest results with our system were reported in [11], but since then some improvements have been made. Cepstral mean subtraction is now used by default and for all features. If statistical significance is mentioned, it is referred to the Wilcoxon signed-rank test with significance level 0.05.

As discussed, the motivation behind constraining the exponential parameters to a subspace is that decreasing the number of parameters decreases the computational cost almost with the same ratio. The number of floating point operations per an input feature for evaluating all the Gaussians is roughly twice the number of parameters for every considered model.

3.2. Finnish LVCSR task

The training data for the Finnish task contains both read and spontaneous sentences and word sequences from 207 speakers, with total of 21 hours of speech. The training set is quite small and might be prone to overfitting so the state tying was tightened to ensure at least 2000 feature vectors per state which resulted in a total of 1602 tied states. As Finnish is a highly-inflectional language, the language modeling is based on morphs learned unsupervisedly from the data as in [13]. The N-grams were trained to varying lengths as in [12]. Because of this highly-inflectional nature, we prefer analyzing letter error rates (LER) instead of word error rates (WER) and WER is given here only as reference.

Table 1: 8 Gaussians per state in the Finnish task

Model	D	LER	WER	#Params
Diagonal	-	4.50	16.12	1.01 M
Full	-	4.08	14.98	10.51 M
PCGMM	40	4.25	15.62	1.06 M
	80	4.00	15.11	1.60 M
	120	**3.85**	14.69	2.14 M
SCGMM	40	5.09	17.62	0.56 M
	80	**4.33**	16.21	1.10 M
	120	4.33	16.28	1.65 M
	160	4.40	16.32	2.19 M

The error rates using 8 Gaussians per tied state are shown in table 1. We note that the full covariance model gives in this setup a relative improvement of 9% over the diagonal model. The error rates with 16 Gaussians per tied state are shown in table 2. Diagonal results naturally improve, but with the full covariance model the training data isn't enough anymore and results in badly overtrained models. The subspace constrained models on the other hand improve over the corresponding 8 Gaussian models so we clearly avoid the overtraining issues.

An interesting comparison can be made between Diagonal ($16G$) and PCGMM ($8G$,$D = 120$) where we note a statistically significant 10% percent relative improvement with roughly the same model cost. Both PCGMM ($8G$, $D = 40$) and SCGMM ($8G$,$D = 80$) give roughly the same performance as the diagonal model but with a halved model cost.

Table 2: 16 Gaussians per state in the Finnish task

Model	D	LER	WER	#Params
Diagonal	-	4.26	15.66	2.03 M
Full	-	4.96	16.58	21.02 M
PCGMM	40	4.06	15.21	2.08 M
	80	**3.77**	14.56	3.14 M
	120	3.83	14.62	4.20 M
SCGMM	40	4.48	16.39	1.08 M
	80	**4.10**	15.33	2.14 M

The PCGMM results typically improve when the basis dimensionality grows. For the SCGMM we get slightly weird behavior which can be seen from the table 1, where at some point the performance starts to degrade. It is hard to say anything certain about this, but the coupling of the first and second order terms to the same space might give for some Gaussians more possibilities for overtraining than it helps in the modeling, because the subspace is shared by all the Gaussians. It was ensured that the likelihoods grow as they should when increasing the basis dimensionality.

Outside the tables, the Diagonal ($32G$) model results in LER 3.89 with the model cost of $4.05M$. Again PCGMM ($8G, D = 120$) gives roughly the same result with a halved model cost.

3.3. English LVCSR task

For training the acoustic models for English we used the Fisher corpus. A 180 hour subset of the corpus was selected with good coverage of different dialects and genders. The state tying was done to ensure at least 4000 feature vectors per tied state which resulted in a total of 5435 tied states. Language modeling was based on words and the N-grams were trained to varying lengths as in [12]. Evaluation is performed with TDT4: Voice of America broadcast news task [2]. Analysis is based on WER.

The results for this task are shown in table 3. We note again that diagonal models improve when increasing the number of Gaussians. Our full covariance evaluations haven't been optimized and were getting too heavy so we don't provide them here. It was decided to fix the subspace dimensionality and try two different number of Gaussians for both PCGMM and SCGMM.

Here the diagonal models and SCGMM models with the same number of Gaussians have comparable complexity. With 16 Gaussians we record a relative improvement of 2%. The differences between diagonal and SCGMM models aren't however statistically significant.

PCGMM performs slightly better and with the same number of Gaussians we note statistically significant improvements over the diagonal models, although with a bigger model cost. Perhaps the most interesting comparisons can be made with the diagonal models that have twice the number of Gaussians than with the PCGMM models. Comparing the PCGMM ($G = 8$,$D = 80$) and Diagonal ($16G$) there is a 23% decrease in the model complexity and correspondingly for PCGMM ($G = 16$,$D = 80$) and Diagonal ($32G$) there is a 24% decrease. The differences between these pairs aren't statistically significant and so we conclude that this decrease comes 'for free'.

Table 3: Results for the English TDT4:VOA task

Model	G	LER	WER	#Params
Diagonal	8	18.75	33.82	3.44 M
	16	17.80	32.53	6.88 M
	32	17.02	31.31	13.74 M
SCGMM D=80	8	18.37	33.90	3.59 M
	16	17.34	**31.90**	7.11 M
PCGMM D=80	8	17.55	32.43	5.28 M
	16	16.74	**30.99**	10.50 M

4. Conclusions

Subspace constrained Gaussian mixture models have been in the recent years used successfully in different speech recognizers. PCGMM has been shown to give a good compression for full covariance models [4] and on the other hand to exhibit some kind of 'smoothing behavior' [10]. The more general SCGMM is also interesting and has been shown to surpass PCGMM in some restricted grammar tasks [3].

The previous tests have been performed in relatively high-quality setups with hundreds of hours of training data. Such databases aren't, however, available for all smaller languages or special tasks. In this paper the models were tried in a more modest system, which puts the robustness of the models to a strict test. It is hard to achieve reasonable full covariance performance in our Finnish task without extensive tweaking.

PCGMM was found to work very well and the results seem to confirm the findings of [10]. To our knowledge the first results applying SCGMM to a LVCSR task were presented here. The results improve in some cases over our baseline diagonal GMM, but perform still typically worse than the PCGMM model. It wasn't also quite as robust as PCGMM in the low-data situation.

5. Acknowledgments

This work was supported by the Academy of Finland in the projects *Adaptive Informatics* and *New adaptive and learning methods in speech recognition*.

6. References

[1] Hilbert Class Library, C++ optimization library, http://www.trip.caam.rice.edu/txt/hcldoc/html/.

[2] Linguistic Data Consortium, http://www.ldc.upenn.edu/.

[3] S. Axelrod, V. Goel, R. A. Gopinath, P.A. Olsen, and K. Visweswariah. Subspace constrained Gaussian mixture models for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 13(6):1144–1160, 2005.

[4] S. Axelrod, R. Gopinath, and P. Olsen. Modeling with a subspace constraint on inverse covariance matrices. In *Proc. of the INTERSPEECH*, pages 2177–2180, 2002.

[5] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical report, University of Cambridge, 1997.

[6] J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, 1995.

[7] P. A. Olsen and K. Visweswariah. Fast clustering of Gaussians and the virtue of representing Gaussians in exponential model format. In *Proc. of the INTERSPEECH*, pages 673–676, 2004.

[8] P.A. Olsen and R.A. Gopinath. Modeling inverse covariance matrices by basis expansion. *Speech and Audio Processing, IEEE Transactions on*, 12(1):37–46, 2004.

[9] P.A. Olsen, K. Visweswariah, and R. Gopinath. Initializing subspace constrained Gaussian mixture models. In *Proc. of the ICASSP*, volume 1, pages 661–664, 2005.

[10] D. Povey. SPAM and full covariance for speech recognition. In *Proc. of the INTERSPEECH*, 2006.

[11] J. Pylkkönen. LDA based feature estimation methods for LVCSR. In *Proc. of the INTERSPEECH*, 2006.

[12] V. Siivola and B. Pellom. Growing an n-gram model. In *Proc. of the INTERSPEECH*, pages 1309–1312, 2005.

[13] T.Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S.Virpioja, and J. Pylkkönen. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech and Language*, 20(4):515–541, October 2006.

[14] V. Vanhoucke and A. Sankar. Mixtures of inverse covariances. *Speech and Audio Processing, IEEE Transactions on*, 12(3):250–264, 2004.