Comparison of ML, MAP, and VB based Acoustic Models in Large Vocabulary Speech Recognition

Panu Somervuo

Neural Networks Research Centre
Helsinki University of Technology
P.O.Box 5400, FI-02015 HUT, Finland
panu.somervuo@hut.fi

Abstract

The present work compares three different methods for training acoustic models in a Finnish large vocabulary speech recognition system. The models are trained using the maximum likelihood (ML), maximum a posteriori (MAP), and variational Bayesian (VB) principle. The results show that when the model complexity is properly chosen, all three methods give similar performance. As the model complexity increases, the performance of ML based system starts to degrade whereas no overfitting is observed using MAP and VB based models. MAP gives slightly better recognition accuracy over VB but it cannot be used for model selection without auxiliary data. The advantage of VB is that it can be used for selecting a well performing model structure using only training data.

1. Introduction

Most speech recognition systems are based on hidden Markov models (HMMs) where the state observation probabilities are modeled by mixture densities. The present work focuses on the estimation of the parameters of Gaussian mixture models (GMMs) of HMM states. Three methods are compared and their effect to speech recognition accuracy is investigated. The methods are based on maximum likelihood (ML) principle, maximum a posteriori (MAP) principle, and variational Bayesian (VB) approach. The asymptotic behavior of all three methods is the same, i.e. they will give the same solution in the limit of infinite amount of training data. Their main difference is how they deal with insufficient amount of training data.

The main interest in practice is to get robust parameter estimates when little or limited amount training data is available. Another interest is to have a tool for selecting the proper model structure, e.g. the optimal number of Gaussians in the mixture model or determining the model complexity for shared-state triphone HMMs. Neither ML nor MAP provides any principled way to do the model selection, but the methods based on variational Bayesian approach can be used for this.

In this study, ML, MAP, and VB based Gaussian mixture model solutions are experimentally compared on the basis of recognition accuracy using a Finnish large vocabulary speech recognition system. The acoustic models of the recognizer are based on triphone HMMs. Several recognizers are trained with varying complexity. Triphone models are clustered using the decision trees with different sizes and the effect of varying the number of Gaussians in the mixture models is investigated.

2. On parameter estimation

Basic ML solution without any regularization has the problem of overfitting the model parameters to the training data. This happens if the model is too complex, i.e. has too many free parameters compared to the amount of training data. In practice some smoothing is done, e.g. a floor value is set for variance estimations in order to avoid singular solutions and smoothen peaky mixture models. Also several parameters can be tied so that the effective number of the free parameters is reduced. But it is not trivial to determine the proper smoothing and amount of parameter tying. In practice, one is forced to divide the original training data into two sets, one used for training the parameters and the other for validating the trained model.

In MAP estimation, prior distributions are defined for the parameters which helps avoiding the overfitting. But MAP is based on the single point of the posterior distribution of model parameters. A different approach is to use Bayesian modeling where the entire posterior distribution is considered [1]. The unknown parameters are integrated out instead of selecting the values based on their point densities. The integration process is called marginalization. Marginalization prevents overfitting and enables to do the model comparison based on the model evidence. The evidence in Bayesian modeling is the marginal likelihood of the data, not the single point value of the density function like in ML and MAP.

Since in practice the real posterior distribution is not available, one must use some kind of approximation for it. One approach is to use Laplace method, where a single Gaussian is fitted to the posterior distribution around the MAP solution. But a more advanced method is to use variational Bayesian (VB) approach [2]. The term variational means that the exact functional form of the posterior approximation is not fixed but the best possible approximation is sought using the calculus of variations. The constraint in [2] is that the posterior approximation must be written in the factorized form.

There is still another type of method for getting the posterior distribution. If samples are taken from the real posterior distribution no approximations are made. But the drawback is that the computation time can be long and in practice it is very difficult to know when the method has been converged so that all relevant parts of the posterior distribution have been sampled. Therefore VB seems to be the most practical approach for applying the Bayesian modeling to the speech recognition systems which typically have large number of parameters.

It should be noted that there exist also several other optimization criteria for training the models than the three methods used in the present study. The fundamental dichotomy between

the training methods can be made between the maximum likelihood and discriminative training approaches. Since the main goal in speech recognition is to do classification from the acoustic feature vector stream into word sequence and maximum likelihood based methods are optimal in classification only if the model is correct and the amount of training data is infinite (which in speech recognition unfortunately is never the case), the discriminative methods should be a better choice. But even though the discriminatively trained models are better in principle, in practice they also have problems since they are even more sensitive to overfitting than ML based models. The question about parameter smoothing and proper model complexity selection is thus an important issue regardless of the parameter optimization criterion. In the current work the focus was on ML, MAP, and VB methods.

3. Recognition system

The recognition system used in the current study is based on triphone HMMs where states are modeled by Gaussian mixture densities. The building blocks of the trigram language model are 20.000 morpheme-like subword units which have been automatically extracted from a large text corpus using unsupervised learning [3, 4]. Recognition output is obtained by time-synchronous stack decoding.

Acoustic data was obtained from a professional female speaker reading a Finnish book in a quiet environment. Training data consisted of 10 hours speech, 10 minutes development data were used for setting the language model weight and the final test data set was 40 minutes.

Speech was sampled at 16 kHz rate. Feature vectors were 24-dimensional, 12 static MFCC features concatenated by 12 delta features. They were computed at 10 ms intervals from 25 ms time windows using HTK software [5]. Feature vectors were scaled component wise to zero-mean and unit-variance using the sample mean and sample variance of the training data.

3.1. Evaluating recognition accuracy

Finnish is agglutinative, compounding language which means that words can be constructed by concatenation. Many prepositions of English language are used as corresponding word suffixes in Finnish. As an example, an English word sequence *also in my home* would be written as one Finnish word *kodi+ssa+ni+kin*. For this reason measuring the recognition accuracy based on the full-word accuracy might be misleading. For example, if the word *my* were recognized incorrectly in the English word sequence, the error rate would be 25%, but if the corresponding Finnish subword *ni* were wrong, the recognition output being e.g. *kodi+ssa+mme+kin*, the error rate would be 100%. Therefore, a better describing measure of recognition accuracy for Finnish language is to use letter error rate [3]. In the above example this would be 3/12 = 25%.

4. Experiments

4.1. Data segmentation

In order to segment the acoustic data into triphone units, acoustic models were first trained using HTK tools following the guidelines in the HTK manual [5]. First monophone HMMs were trained, a three-state HMM for each phone using a single Gaussian for each state. Triphone models were constructed by copying the monophone models into triphone models and then clustering the states of the resulting models using the decision tree approach. Two cycles of Baum-Welch re-estimation were performed between each step. The questions for the decision tree were designed using knowledge about Finnish phonetics.

Two decision trees were constructed with different threshold values for stopping the node splitting. The first tree where the tying parameters in the HTK's HHEd command script were `RO=100.0` and `TB=350.0` resulted in 2748 shared states. Another tree with the parameters `RO=10.0`, `TB=35.0` resulted in 7247 shared states. The HTK based models were then used for segmenting the training data into triphone-state specific segments. This fixed data segmentation was used in the following experiments where different parameter estimation methods where compared for constructing the Gaussian mixture models of the HMM states. Transition probabilities between the states were also kept fixed.

4.2. Mixture model initialization

Mixture models were trained separately for each HMM state using the fixed data segmentation. The same initializations were used for ML, MAP, and VB algorithms. The initial Gaussian means were obtained from the vector quantization process. Initial code vectors were randomly picked from the appropriate data segments and the batch-mode Self-Organizing Map (SOM) [6] based initialization was performed for each state specific codebook. Batch-SOM is like k-means algorithm where the training data is smoothly shared between the code vectors by means of the neighborhood function. During the codebook training, the width of the Gaussian-shaped neighborhood function was smoothly decreased to zero so that in the end of the codebook training the algorithm behaved like k-means algorithm. The use of the neighborhood in the SOM training has the effect to help the k-means algorithm to escape from possibly bad initialization and local minima.

4.3. Mixture model training

The Gaussian mixture model with K mixture components is:

$$p(\mathbf{y}) = \sum_{s=1}^{K} \omega_s \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s) \tag{1}$$

where $\mathbf{y}$ is the data vector, ω_s denotes the mixture weight, $\boldsymbol{\mu}_s$ mean of Gaussian and $\boldsymbol{\Sigma}_s$ the covariance matrix. Diagonal covariance matrices were used.

Each shared state specific mixture model was trained using those feature vectors which were mapped to it in the previous HTK based segmentation. So instead of Baum-Welch training, Viterbi-type training was now used with fixed segment borders. Training of each GMM was iterated till the relative change in the cost function was below 10^{-5} or the maximum number of training cycles (ten) was exceeded. For notational simplicity, the update formulas are presented for one mixture model without any state indices, N denotes the number of feature vectors mapped to it and d is the dimension of the feature vector.

The ML updates, based on expectation maximization (EM) algorithm, are:

$$\begin{aligned} \hat{\omega}_s &= \frac{\sum_{n=1}^{N} \zeta_s^n}{\sum_{s=1}^{K}\sum_{n=1}^{N} \zeta_s^n}, \qquad \hat{\boldsymbol{\mu}}_s = \frac{\sum_{n=1}^{N} \zeta_s^n \mathbf{y}_n}{\sum_{n=1}^{N} \zeta_s^n}, \\ \hat{\boldsymbol{\Sigma}}_s &= \frac{\sum_{n=1}^{N} \zeta_s^n (\mathbf{y}_n - \hat{\boldsymbol{\mu}}_s)(\mathbf{y}_n - \hat{\boldsymbol{\mu}}_s)^T}{\sum_{n=1}^{N} \zeta_s^n}, \end{aligned} \tag{2}$$

where ζ_s^n is the responsibility of mixture component s for generating the feature vector $\mathbf{y}_n$:

$$\zeta_s^n \propto (2\pi)^{-\frac{d}{2}} |\mathbf{\Sigma}_s|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{y}_n - \boldsymbol{\mu}_s)^T \mathbf{\Sigma}_s^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_s)} . \quad (3)$$

The MAP estimates are [7]:

$$\hat{\omega}_s = \frac{(\lambda^0 - 1) + \sum_{n=1}^N \zeta_s^n}{\sum_{s=1}^K (\lambda^0 - 1) + \sum_{n=1}^N \zeta_s^n} , \hat{\boldsymbol{\mu}}_s = \frac{\beta^0 \boldsymbol{\rho}^0 + \sum_{n=1}^N \zeta_s^n \mathbf{y}_n}{\beta^0 + \sum_{n=1}^N \zeta_s^n}$$
$$\hat{\mathbf{\Sigma}}_s = \frac{\mathbf{\Phi}^0 + \beta^0 (\boldsymbol{\rho}^0 - \hat{\boldsymbol{\mu}}_s)(\boldsymbol{\rho}^0 - \hat{\boldsymbol{\mu}}_s)^T + \sum_{n=1}^N \zeta_s^n (\mathbf{y}_n - \hat{\boldsymbol{\mu}}_s)(\mathbf{y}_n - \hat{\boldsymbol{\mu}}_s)^T}{(\nu^0 - d) + \sum_{n=1}^N \zeta_s^n} \quad (4)$$

where $\lambda^0, \boldsymbol{\rho}^0, \beta^0, \nu^0$, and $\mathbf{\Phi}^0$ are the prior parameters. Parameters $\{\omega_s, \boldsymbol{\mu}_s, \mathbf{\Sigma}_s\}$ have the following conjugate priors: mixture weights are jointly Dirichlet, $\{\omega_s\} \sim \mathcal{D}(\lambda^0)$, the means conditioned on the inverse covariance matrices are Normal, $\boldsymbol{\mu}_s | \mathbf{\Sigma}_s^{-1} \sim \mathcal{N}(\boldsymbol{\rho}^0, \frac{1}{\beta^0} \mathbf{\Sigma}_s)$, and the inverse covariance matrices are Wishart. In the present work where diagonal covariance matrices were used, the diagonal elements have Gamma priors, $p(\mathbf{\Sigma}_s^{-1}) = \prod_{i=1}^d \mathcal{G}(\nu^0, \mathbf{\Phi}_i^0)$. The following values were used: $\lambda^0 = 1$, $\beta^0 = 1$, $\nu^0 = 1$. The priors of the Gaussian means for triphone state GMMs were defined so that $\boldsymbol{\rho}^0$ and $\mathbf{\Phi}^0$ were the mean and the covariance matrix of the segmented training data based on the three-state monophone HMM. Monophone models were thus used as priors for triphone models.

In the VB method, the same priors were used as in the MAP estimation. The update algorithm [2] is similar to the MAP estimation, but instead of maximizing the auxiliary function of EM algorithm by seeking the values of parameters based on the single points of the likelihood function, the unknown parameters are integrated out over the variational posterior approximation.

The parameter posterior is computed in two stages, first the parameters $\{\omega_s, \boldsymbol{\mu}_s, \mathbf{\Sigma}_s\}$ are updated:

$$\bar{\omega}_s = \frac{1}{N} \sum_{n=1}^N \gamma_s^n , \qquad \bar{\boldsymbol{\mu}}_s = \frac{1}{\bar{N}_s} \sum_{n=1}^N \gamma_s^n \mathbf{y}_n ,$$
$$\bar{\mathbf{\Sigma}}_s = \frac{1}{\bar{N}_s} \sum_{n=1}^N \gamma_s^n (\mathbf{y}_n - \bar{\boldsymbol{\mu}}_s)(\mathbf{y}_n - \bar{\boldsymbol{\mu}}_s)^T , \quad (5)$$

where $\bar{N}_s = N \bar{\omega}_s$, and γ_s^n is the responsibility of mixture component s for generating the observation $\mathbf{y}_n$. This expression is slightly more complicated than (3) and the details can be found in [2, 8]. The posterior parameter updates are:

$$\lambda_s = \bar{N}_s + \lambda^0 , \qquad \boldsymbol{\rho}_s = (\bar{N}_s \bar{\boldsymbol{\mu}}_s + \beta^0 \boldsymbol{\rho}^0)/(\bar{N}_s + \beta^0) ,$$
$$\beta_s = \bar{N}_s + \beta^0 , \qquad \nu_s = \bar{N}_s + \nu^0 ,$$
$$\mathbf{\Phi}_s = \bar{N}_s \bar{\mathbf{\Sigma}}_s + \bar{N}_s \beta^0 (\bar{\boldsymbol{\mu}}_s - \boldsymbol{\rho}^0)(\bar{\boldsymbol{\mu}}_s - \boldsymbol{\rho}^0)^T / (\bar{N}_s + \beta^0) + \mathbf{\Phi}^0 \quad (6)$$

The predictive density based on the final values of the posterior parameters is obtained by integrating out the parameters. The result is a mixture of Student-t distributions [2]. This was used for computing the VB state likelihoods in the recognition experiments.

5. Results

The number of the components in the Gaussian mixture densities were varied and its effect to the recognition accuracy was investigated using ML, MAP, and VB based estimation methods. The results in Table 1 are for development set which was used for tuning the language model weights separately for each recognition system. The number of the shared states is the result of the decision tree based clustering and a Gaussian mixture model was trained for each shared state. When the number of the models is 2748 there is no overfitting observed in ML models even when 20 Gaussians per state is used and all three methods perform equally well.

Table 1: *Recognition results for 10-minute development set (603 words, 5380 letters). Language model weights were tuned to minimize the letter error rate. Two underlined rows are for the VB system with unequal number of mixture components per GMM based on the VB cost function.*

	Letter Error Rate			Word Error Rate		
#Gaussians	ML	MAP	VB	ML	MAP	VB
2748 × 1	4.8	4.8	5.0	25.0	25.2	26.4
2	4.1	4.1	4.1	23.4	22.7	23.1
5	3.2	3.4	3.3	19.2	20.6	20.2
10	3.3	3.3	3.4	19.9	20.2	20.3
15	3.1	3.1	3.2	19.4	19.4	18.7
20	3.0	3.1	3.1	18.1	19.1	18.7
ave 10.4			3.2			19.1
7247 × 1	5.1	5.1	5.2	26.9	25.2	25.9
2	4.2	4.1	4.3	24.7	23.4	23.9
5	4.3	3.7	4.1	24.9	20.9	21.6
10	5.1	3.1	4.1	29.0	18.1	22.1
15	5.8	3.1	3.8	32.7	19.1	21.4
ave 4.9			4.1			22.2

Table 2: *Recognition results for 40-minute test set (2621 words, 23200 letters) using 2748- and 7247-GMM systems. VB systems had unequal number of Gaussians per GMM, on average 10.4 and 4.9.*

	Letter Error Rate			Word Error Rate		
#Gaussians	ML	MAP	VB	ML	MAP	VB
2748 × 10	3.4	3.5	3.3	20.3	21.71	19.7
7247 × 5	5.1	3.9	4.6	28.7	23.2	23.3

In the second system with 7247 models when sufficiently small number of Gaussians per mixture was used, all methods performed again equally well. But after increasing the mixture model size to 10, ML starts to overfit. MAP continues to increase the recognition accuracy whereas the performance of VB based models seems to saturate. Since same priors are used for MAP and VB the main difference lies in the averaging. Based on [2], the predictive density for VB based GMM is the mixture of Student-t distributions. When the amount of training data is large, Student-t distribution is close to Gaussian, but the differences are most noticeable using small amount of training data. VB based models may give better likelihood to data, but here MAP based models succeeded to discriminate the data better.

Two rows in Table 1 are for VB based system where unequal number of Gaussians were used for mixture models. The number was determined by the VB cost function, see details in [2, 8]. Because of the embedded penalization for complex models, VB cost function can be used for selecting the best number of Gaussians. This does not necessarily minimize the recognition error, since the objective of VB is to approximate the marginal likelihood of data. Nevertheless, it can be seen that good results were obtained. The average number of Gaussians per mixture model was 10.4 when 2748 GMMs were used

and 4.9 when 7247 GMMs were used. Another way to set unequal number of components to mixture models would be to train GMMs with initially large number of components and then after training prune weakly used components from each model. In MAP and VB, those components should have values close to priors.

Results for test data are in Table 2. The number of Gaussians per state was determined to be 10 and 5 for the two systems with 2748 and 7247 GMMS. In the smaller system all methods performed equally well, but in the larger system MAP gave the best letter error rate VB being the second best.

5.1. VB based state clustering

The two recognizers with 2748 and 7247 states were based on the state clustering using HTK's decision tree algorithm. The tree node splitting was based on the increase of the likelihood of the data. Since the likelihood will increase as the model complexity increases, there must be a manually selected threshold for stopping the tree growing. This threshold clearly affects performance of the ML based models as can be seen when comparing the results in Table 1.

Here it was experimented how the decision tree can be constructed without manually setting the stopping criteria [9, 10]. Since VB framework enables to compute the lower bound of the marginal likelihood of the training data (model evidence), this measure can be used for determining the proper tree structure. For each node, the difference between marginal likelihoods before and after candidate node splitting is computed and if the result is positive, the splitting is done. The lower bound of the marginal likelihood can be interpreted as a penalized likelihood and the system has therefore embedded complexity control.

The VB state clustering did not improve the recognition accuracy obtained by HTK clustering but gave similar results. The number of the nodes in the VB decision tree was 6368. Using one Gaussian per state the letter error rate was 5.1% for development set, this is the average of the VB letter error rates of 2748×1 and 7247×1 Gaussian systems in Table 1.

Although the recognition accuracy was the same as when using HTK based state clustering, the benefit of the VB method is that it does not require any manually set threshold for stopping the node splitting in the construction of the decision tree.

6. Discussion

The main interest in the present work was to compare MAP against VB. ML based models were used more like a baseline system. In the previous works where VB has been applied to speech recognition [9, 10, 11], no comparison to MAP based models have been made. It was a bit surprising to observe that the MAP based models gave better recognition accuracy than VB based models when large number of GMMs was used.

The attractive property of VB method is that its cost function has embedded complexity penalization and it can therefore be used for model selection. But the cost function is the lower bound for the marginal likelihood and in speech recognition the aim is to do classification. The discrimination of phone classes is more important than maximization of the likelihood. Therefore it would be extremely interesting to try to apply VB framework also to discriminative training.

Another interesting topic would be to determine the optimal prior strengths for MAP and VB. In the present work relatively weak priors were used. The comparison between MAP and VB was fair, since the same initializations of the GMMs and equal priors were used for both methods, but the performance of both methods could be improved by different choice of prior strengths.

7. Conclusions

In this work, ML, MAP, and VB based acoustic models were compared in the large vocabulary speech recognition task. Acoustic models were triphone HMMs with decision tree based state clustering. Gaussian mixture model was trained for each state and the number of the mixture components was varied. All three methods gave similar recognition accuracies when sufficiently small number of GMMs were used. When the number of the GMMs increased and the number of the mixture components increased, differences between the methods could be observed. The performance of the ML based models started to degrade whereas MAP and VB based models did not suffer from the overfitting. VB based models performed better than ML based models, but MAP based models gave the best recognition accuracy.

8. Acknowledgements

The author would like to thank Vesa Siivola and Teemu Hirsimäki for providing the language models and the decoder used in the current study and Matias Creutz for designing the phonetic questions in the construction of the decision tree.

9. References

[1] Gelman, A., Carlin, J., Stern, J., Rubin, D., Bayesian Data Analysis, Chapman & Hall, 1995.

[2] Attias, H., "A Variational Bayesian Framework for Graphical Models", Proc. NIPS 12, pp. 209–215, 2000.

[3] Siivola, V., Hirsimäki, T., Creutz, M., Kurimo, M., "Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner", Proc. Eurospeech 2003, pp. 2293-2296.

[4] Creutz, M., Lagus, K., "Unsupervised Discovery of Morphemes", Proc. Workshop on Morphological and Phonological Learning of ACL 2002, pp. 21–30.

[5] Young, S. & al., The HTK Hidden Markov Model Toolkit, <http://htk.eng.cam.ac.uk/>

[6] Kohonen, T., The Self-Organizing Map, Springer, 2001.

[7] Gauvain, J., Lee, C., "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", IEEE Tr. SAP, Vol. 2(2):291–298, 1994.

[8] Somervuo, P., "Speech Modeling using Variational Bayesian Mixture of Gaussians", Proc. ICSLP 2002, pp. 1245–1248.

[9] Watanabe, S., Minami, Y., Nakamura, A., Ueda, N., "Constructing Shared-State Hidden Markov Models based on a Bayesian Approach", Proc. ICSLP 2002, pp. 2669-2672.

[10] Watanabe, S., Minami, Y., Nakamura, A., Ueda, N., "Application of the Variational Bayesian Approach to Speech Recognition", Proc. NIPS 15, pp. 1261–1268, 2002.

[11] Valente, F., Wellekens, C., "Variational Bayesian GMM for Speech Recognition", Proc. Eurospeech 2003, pp. 441–444.