Generating Gaussian Mixture Models by Model Selection For Speech Recognition

Kai Yu
F06 10-701 Final Project Report
kaiy@andrew.cmu.edu

Abstract

While all modern speech recognition systems use Gaussian mixture models, there is no standard method to determine the number of mixture components. Current choices for mixture component numbers are usually arbitrary with little justification. In this paper we apply some common model selection methods to determine the number of mixture components. We show that they are ill-suited for the speech recognition task because increasing test set data likelihood does not necessarily improve speech recognition performance. We then present a model selection criterion modified to better suit speech recognition, and its positive and negative effects on speech recognition performance.

1 Introduction

Modern speech recognition algorithms are filled with data-driven techniques that build models to represent human speech. While specific implementations may differ in the algorithms used, one commonality of all modern speech recognizers is the use of Gaussian mixture models (GMMs) to represent senones, the atomic units of speech. For each time interval, or frame, a feature vector is computed to represent the acoustic information, and this is inputted into the GMMs to compute the probabilities of each senone. These probabilities are then used by the search backend to determine the most likely sequence of sounds, which determines the most likely sequence of words.

Although using GMMs is a standard method, there is no consensus on how to determine the number of mixture components per senone. The most common methods are setting the number of mixture components of a particular senone to an arbitrary constant, or a fraction of instances of that senone in the training set. However, there is no statistical justification for these methods, and neither of these methods take the complexity of the senone acoustic distribution into account. Also, even though the GMMs are generated via expectation maximization, almost no work has been done to cross-validate the results or check for overfitting. While some work has been done to prune mixture components using the Bayesian information criterion [1], these methods only consider the data that belongs to a particular senone when creating the model, when the effects from other senones should be considered. An attempt [2] has been made on trying to include all the data when training the models, but it generates statistically unjustified models.

In this paper we introduce a method to generate GMMs whose number of mixture components are statistically justified, and which take into account effects from other senones. To include information regarding other senones, we add a misclassification parameter so that easily misclassified senones will have more components to make it more distinctive. We also compare the results against common model selection methods, including those used in previous work.

The paper is organized as follows: section 2 describes the model selection methods used and section 3 reports the results from these methods. Section 4 introduces the new method based on the insight derived from the results in section 3. Test results are presented in section 5, and conclusions in section 6.

2 Model selection methods

To determine the number of components per GMM to best represent the true distribution of the data, model selection (MS) techniques will be used. This will provide a method to maximize the likelihood of the training data while attempting to avoid overfitting. Besides using the n-fold cross-validation (CV) technique, where the data is is trained on (n-1) folds and tested on the held out set, the information criterion techniques described below will also be used.

One model selection technique is the Akaike Information Criterion (AIC)[3], which penalizes the model based on its complexity. It is defined as:

$$AIC(\theta) = -2\log p(X|\theta) + 2k \tag{1}$$

where θ is the model, X is the input data, $\log p(X|\theta)$ is the log of the probability of X given θ, and k is the number of parameters in the model θ. The model selected will have the lowest AIC score. However, in cases where the number of input vectors, n, is small relative to k, AIC tends to have a negatively-biased estimate of the difference between the model and true distribution. To account for this, a corrected AIC (AICc)[4] has been previously proposed, which is defined as:

$$AICc(\theta) = -2\log p(X|\theta) + 2k\left(\frac{n}{n-k-1}\right) \tag{2}$$

When the number of samples and the complexity of the model are of the same order of magnitude, the penalty term for AICc will be much larger than AIC, which corrects for the negative bias. As the number of samples becomes much larger than the model complexity, AICc will converge to AIC. Again the model that minimizes AICc will be chosen, and AICc tends to perform better than AIC when the number of samples is limited.

The final model selection method that will be used is the Bayes Information Criterion (BIC)[5], defined as:

$$BIC(\theta) = -2logp(X|\theta) + k\log(n) \tag{3}$$

The model that minimizes BIC will be selected.

3 Model selection results

The following experiments are performed using the Communicator corpora [6], which consists of telephone calls for air travel planning. It contains 2165 senones, and a training set of about 3,000,000 frames and test set of about 300,000 frames were used. The complete Communicator training set is approximately 10 times larger, but due to time and resource constraints only a subset is used. The input feature vector has 39 elements generated from mel frequency cepstral coefficients [7] and its first and second time derivatives. To classify each frame, existing GMMs trained using Sphinx-Train[8] on the entire Communicator training set are used to force-align the waveform and the transcript to find the optimal sequence of senones. The speech recognizer we use for force-alignment and decoding is Sphinx 3.0 [8]. To measure speech recognition performance, we use accuracy, which is

$$\text{Accuracy} = 100 \cdot \frac{\text{Number of reference words correctly decoded in hypothesis}}{\text{Number of reference words}}, \tag{4}$$

and word error rate (WER), which is

$$\text{WER} = 100 \cdot \frac{\text{Number of substitutions + deletions + insertions in hypothesis}}{\text{Number of reference words}}. \tag{5}$$

Accuracy favors sentences with correctly decoded words, while WER is biased towards sentences with less hypothesized words. WER and a comparison of the two metrics are explained in more detail in the appendix. Increasing accuracy or decreasing WER corresponds to better speech recognition performance.

GMMs are created by taking the all the frames classified to a particular senone and running it through Cluster [9], which has been modified to initialize the mixture components randomly instead of deterministically. We arbitrarily restrict the maximum number of components per GMM to 32 and diagonal covariance matrices for the Gaussians due to time constraints. The maximum number of GMM components for senone i, N_i, is set to

$$N_i = \min\left(32, \frac{\text{Number of frames classified as } i}{39}\right). \tag{6}$$

To make sure the Gaussians were not ill-formed, we require at least one frame per Gaussian dimension. For senones that appeared in less than 39 frames, we use the single component Gaussian model generated by Sphinx-Train. Ideally, for each number of mixture components we would like to train multiple GMMs to try to find the global optimum, but this process is too time-consuming. Instead, a GMM for senone i with N_i mixture components is first created, and then mixture components whose combination resulted in the smallest change in data likelihood are merged to initialize the clustering algorithm to create a GMM with $N_i - 1$ mixture components. This process is repeated until only one component remained, and number of components for the GMM with the lowest information criterion is selected. Since only Gaussians with diagonal covariance matrices are considered, the number of parameters per mixture component was 79 (1 prior + 39 means + 39 variances).

To verify that the models trained using Cluster could be plugged into Sphinx, a 4-mixture component GMM was trained per senone by Cluster and compared against the corresponding model trained by Sphinx-Train. As seen in Table 1, in terms of both accuracy and WER Cluster performed better than Sphinx on the training set, but worse of the test set. This is expected because the Sphinx-Train model is trained on the complete Communicator training set while the Cluster model is trained on the reduced training set, so the Sphinx-Train model should generalize better to the test set and the Cluster model do better just on the data it was trained on. However, since the performance of the Cluster models is comparable to Sphinx-Train, the Cluster models can be used in Sphinx.

Table 1: Results of 4-mixture component GMMs trained by Sphinx-Train and Cluster

	Sphinx-Train	**Cluster**
Training Set Accuracy	81.9	82.5
Training Set WER	30.5	26.8
Test Set Accuracy	85.5	80.0
Test Set WER	27.3	33.8

Table 2: Results of models trained using different model selection methods

	No MS	**AIC**	**AICc**	**BIC**	**5-fold CV**
Average Number of Components	15.68	14.09	**4.96**	5.62	13.47
Training Set Accuracy	**96.7**	96.6	93.7	94.6	96.0
Training Set WER	**14.9**	15.1	19.6	18.4	16.3
Test Set Accuracy	84.9	**85.2**	84.0	84.7	84.7
Test Set WER	22.4	**22.2**	24.8	23.9	23.8
Test Set Likelihood (Normalized)	1.00	**1.03**	0.987	0.991	1.02

Next we generated models using Cluster and applied the model selection methods presented in section 2, and display the results in Table 2. For every row the best score is highlighted in bold, and note that the test set data likelihood is normalized relative to the model with before any model selection has applied.

We note these 5 interesting observations:

- As the average number of components decreases, speech recognition performance on the training set also decreases, and speech recognition performance on the test set also decreases except for AIC.
- Only AIC and CV increased the test set likelihood over the original model with no model selection applied. There may not have been enough training data to fully represent the complexity of some of the senones, causing AICc and BIC to be overly aggressive in pruning the number of mixture components.
- Increased test set likelihood does not necessarily mean better speech recognition performance. CV almost has the highest test set likelihood, but its speech recognition performance is closer to BIC which has the fourth highest test set likelihood.
- Model selection can be beneficial, for using AIC results in the model with the highest accuracy and lowest WER. It also has the highest test data likelihood, but only slightly decreases the average number of components.
- In the models trained using 5-fold CV, for every senone the number of mixture components that maximized the data likelihood was the maximum number of mixture components that could be trained. This number can be less than N_i, because the training set is reduced by the size of the held-out fold. This effect will be lessened if more folds of a smaller size were used to increase the training set size. However, this would be too time-consuming because the 5-fold CV already takes well over a week to run.

Although these results suggest that maximizing test set likelihood corresponds to better speech recognition performance, the 5-fold CV results show there is a small difference. For good speech recognition performance, what matters is not the test set likelihood, but rather the difference in likelihood between the correctly decoded sentence and the most probable incorrect sentence. For example, having the correct sentence and best incorrect sentence likelihood of 0.02 and 0.01 is desirable because the correct sentence will be decoded due to its higher likelihood. If the correct and incorrect sentence likelihoods are 0.5 and 0.7, while the correct sentence likelihood may be higher than the previous example, the incorrect sentence will be decoded making speech recognition performance worse.

The ideal model selection method would have the speech recognition performance of AIC and a reduced number of mixture components similar to AICc or BIC. Since AIC tends to overestimate the number of mixture components, it suggests that a model with a smaller average number of mixture components may be able to perform just as well. While AICc and

BIC are clearly too aggressive, with some modifications designed specifically for speech recognition they may achieve speech recognition performance comparable to AIC while still averaging much fewer mixture components.

4 Proposed model selection criterion

Since the results show that model selection can be beneficial, a model selection criterion modified to better match speech recognition may produce a better model. Ideally the modified model selection criterion would directly optimize speech recognition performance, but it is difficult to incorporate the highest likelihood of an incorrect sentence into the model criterion. Instead we focus on a simpler problem, which is adjusting the penalty on complexity based on the number of times a senone is misclassified. The senones that are often misclassified need more resolution (i.e. mixture components) in order to be represented more accurately and with higher probability, which increases the likelihood of the correct sentence.

To find the number of times a senone is misclassified, we take all the training sentences that had been incorrectly decoded, and compare its sequence of senones to the optimal sequence of senones for the correct sentence. For the frames where the incorrect sentence's senone has a higher likelihood, the correct sentence's senone gets a false negative, and the incorrect sentence's senone gets a false positive. We define the number of times senone i is misclassified per instance, E_i, as

$$E_i = \frac{\text{\# false negatives of } i + \text{\# false positives of } i}{\text{\# training frames that should be classified as } i}. \tag{7}$$

The higher E_i is, the more mixture components are needed to represent senone i. Including the number of false negatives is obvious because with more mixture components the likelihood of all instances of senone i should increase. We also include the number of false positives because with more mixture components senone i should be misclassified less often. We normalize by the number of frames that should be classified as i because we do not want to be biased to senones that appear often.

We modify BIC to create a modified BIC (mBIC)

$$mBIC(\theta) = -2logp(X|\theta) + k\log(n) * \lambda \tag{8}$$

where λ is the a function of E_i. For senones that have large E_i, λ should be 0 in order to lessen the penalty on complexity and increase the number of mixture components. Similarly, for senones with small E_i, λ should be close to 1 because the original BIC worked well. Thus, we generate the following equation for λ:

$$\lambda_i = 1 - \min(1, E_i) \tag{9}$$

Because E_i can be greater than 1 when senone i has many false positives, we need to add the $\min$ term for λ.

5 Proposed model selection criterion results

To find the E_i values, we used the model generated by Cluster with no model selection applied and ran speech recognition on the 3,000,000 frame training set. 21% of the training set sentences were incorrectly decoded, and 19% of the frames of the incorrectly decoded sentences were misclassified, which yielded a total of 184,536 misclassified frames. As seen in Figure 1, most of the misclassifications can be attributed to less than 10% of the senones which are mostly related to senones representing noise, silence, or a variant of the

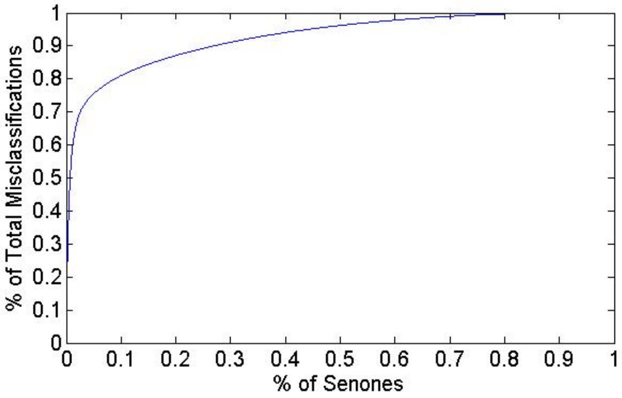

Figure 1: Plot of % of total misclassifications versus % of senones

vowel ”a”. For the noise-related and silence-related senones with lots of misclassifications, the maximum number of mixture components had already been used to represent them, so the modified model selection criterion would not help.

After calculating E_i, we then applied mBIC to the test set and compared it against AIC and BIC in Table 3. Not only does mBIC improve upon BIC, it also has higher test set accuracy than AIC while using less than half of the components. However, its speech recognition performance is not conclusively better than AIC because its WER is higher. Looking at the decoded sentences using AIC and mBIC shows that mBIC often decodes nonsense words where AIC decodes noise, which causes the WER to increase but leave the accuracy unchanged because noise is not considered a reference word. mBIC probably still underestimates the number of parameters needed to represent noise, leading to its higher WER. mBIC has the smallest test set data likelihood of all the model selection methods used, which further highlights that increasing test set data likelihood does not necessarily improve speech recognition performance.

Table 3: Results of models trained using mBIC compared against AIC and BIC

	mBIC	**AIC**	**BIC**
Average Number of Components	6.28	14.09	5.62
Training Set Accuracy	94.9	96.6	94.6
Training Set WER	17.8	15.1	18.4
Test Set Accuracy	85.3	85.2	84.7
Test Set WER	23.3	22.2	23.9
Test Set Likelihood (Normalized)	0.981	1.03	0.991

6 Conclusions

In this paper we have shown that standard model selection techniques that improve test set data likelihood do not necessarily improve speech recognition performance. We also introduce a modified version of BIC designed specifically for speech recognition that can create GMMs with 60% fewer components than the model with no model selection applied while achieving higher accuracy. However, it is not clearly better since it performs worse when using WER as the metric. The model generated with mBIC also is quite comparable to the 4-mixture component GMM model trained by Sphinx-Train. With only 1/10 the training data, it achieves a lower WER and similar accuracy without increasing the average number of components by too much.

For future work, there will not be no time constraint, so we would like to address the limitations of the compromises made due to a lack of time. First, we would like to train the models using the entire training set, which will also enable us to make better comparisons with the models trained by Sphinx-Train. We would like to not restrict the maximum number of mixture components per GMM to 32, because the results also show that some senones need more components to fully express its complexity. We could also try other model selection techniques such as the Dirichlet process and see how it performs, and see if we can gain more insight to help us better modify mBIC to perform even better.

Appendix

Given a reference (correct) sentence and a hypothesis (decoded) sentence, the word error rate (WER) is defined as

$$\text{WER} = 100 * \frac{\text{Number of substitutions + deletions + insertions in hypothesis}}{\text{Number of reference words}}.$$

A deletion occurs when a word in the reference sentence does not exist in the hypothesis sentence. A substitution occurs when a word in the reference is decoded into a different word in the hypothesis. An insertion occurs when a word that does not appear in the reference appears in the hypothesis. While there are many different ways to categorize errors into deletions, substitutions, and insertions, the WER is calculated using the categorization that yields the smallest amount of deletions, substitutions, and insertions.

For example, consider the following hypothesis and reference:

REF:	She	is	with	me		.
HYP:	He	is		me	too	.

There are three errors, which can categorized as 1 substitution (he/she), 1 deletion (with), and 1 insertion (too). Alternatively, we can categorize them as 3 substitutions (he/she, me/with, too/me). Either way, there are three errors so the WER is 75%. Any categorization of errors that has more than 3 errors is incorrect.

Accuracy favors hypotheses with correctly decoded words, while WER is biased towards sentences with less incorrectly hypothesized words. Both metrics are commonly used, and both have their own merits. For example, consider these two possible hypotheses:

REF:		Book	me	that	flight	
HYP1:	Please	book		that	flight	okay
HYP2:			Me	very	flight	

Looking at the two sentences, it seems that hypothesis 1 is much better because it conveys the meaning of the reference sentence, while hypothesis 2 looks like nonsense. If we use accuracy, hypothesis 1 does score higher, (75% versus 50%), but it does worse in WER (75% versus 50%)! While in this example accuracy is the preferred measure, here is an example where WER does better:

REF:			Book	me	that	flight
HYP1:			Book		that	flight
HYP2:	Do	not	book	me	that	flight

Hypothesis 2 is obviously worse since it has the opposite meaning of the reference. However, using accuracy hypothesis 2 has 100% accuracy and is better then hypothesis 1 (75%)! WER will choose the preferred hypothesis 1, whose 25% WER is lower than hypothesis 2's 50%.

References

[1] Chen, S.S. & Gopalakrishnan, P.S. Clustering via the Bayesian information criterion with applications in speech recognition. In *Proceedings of ICASSP*, 645-648, 1998.

[2] Padmanabhan, M. & Bahl, L.R. Model complexity adaptation using a discriminant measure *IEEE Transactions on Speech and Audio Processing*, 8(2): 205-208, 2000.

[3] Akaike, H. Information theory and an extension of the maximum likelihood principle *2nd International Symposium on Information Theory* 267-281, 1973.

[4] Hurvich, C.M. & Tsai, C.L. Regression and time series model selection in small samples. *Biometrika* **76** 297-307, 1989.

[5] Schwarz, G. Estimating the dimension of a model *Annals of Statistics* **6** 461-464, 1978.

[6] Bennett, C. & Rudnicky, A.I. The Carnegie Mellon communicator corpus. In *Proceedings of ICSLP*, 341-344, 2002.

[7] Davis, S.B. & Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357-366, 1980.

[8] CMU Sphinx Open Source Speech Recognition Engines. http://cmusphinx.sourceforge.net/html/cmusphinx.php.

[9] Bouman, C.A. Cluster: {A}n unsupervised algorithm for modeling {G}aussian mixtures. http://www.ece.purdue.edu/˜bouman, 1997.