A SEQUENTIAL FEATURE SELECTION ALGORITHM FOR GMM-BASED SPEECH QUALITY ESTIMATION

Tiago H. Falk and Wai-Yip Chan

Department of Electrical and Computer Engineering
Queen’s University, Kingston, Ontario, Canada
Email: \{falkt, chan\}@ee.queensu.ca

ABSTRACT
We propose a sequential feature selection algorithm for designing Gaussian mixture model (GMM) based estimators. Feature selection is performed progressively to minimize estimation errors. The algorithm is applied to design estimators of subjective speech quality. Simulation shows that estimators designed using the proposed algorithm outperform two benchmark algorithms by as much as 39% in correlation and 24% in root-mean-squared error. Furthermore, features selected by the proposed algorithm are suitable for diagonal GMM estimators, which incur lower computational complexity.

1. INTRODUCTION
In regression, given the values of \( n \) predictor variables, the value of a target variable is estimated by means of a mapping from the predictor variables to the target variable. Regression analysis provides a means to find a best mapping in the form of a regression function. This paper focuses on the case where the joint distribution of the predictor and target variables is modelled by a Gaussian mixture model (GMM). A large class of probability densities can be approximated using Gaussian mixtures. Moreover, GMMs provide a closed form expression for the regression function.

Estimation (or regression) using GMMs is introduced in [1] and the idea is used in [2] to adjust the magnitude spectrum of a speech signal when the fundamental frequency of the signal is altered. In [3] GMMs are used to estimate missing line spectral frequencies and in [4], subjective speech quality ratings.

The choice of the predictor or feature variables is often crucial in regression analysis, as redundant or noisy features degrade estimation performance. The problem at hand is to pick \( m \) feature variables out of \( n > m \) variables for the regression function. The best \( m \) is often not known a priori, and an exhaustive search for an optimal feature subset entails examining \( 2^n - 1 \) possible subsets, a clearly impossible task for large \( n \). One approach is to use common feature selection algorithms such as classification and regression trees (CART) [5] and multivariate adaptive regression splines (MARS) [6].

When designing GMM-based estimators, the features selected by the above algorithms may not lead to high estimation accuracy. In [7], the concept of feature saliencies in the context of GMMs is proposed. By adopting a penalty criterion, saliencies of irrelevant features go to zero, thus performing feature selection. This feature selection procedure, however, does not take the GMM regression function into consideration and may still lead to features that are inefficient for the estimation task at hand.

Here, we propose a feature mining algorithm targeted to estimation tasks that make use of GMM regression functions. An experiment consisting of predicting the subjective quality rating of speech is performed and simulation results show that GMM estimators designed using our proposed algorithm outperform estimators trained on features selected by CART or MARS.

2. GAUSSIAN MIXTURE MODELS

2.1 Background
Let \( \mathbf{u} \) be an \( K \)-dimensional vector, a Gaussian mixture density is a weighted sum of \( M \) component densities

\[
p(\mathbf{u} | \mu, \Sigma, \alpha) = \sum_{i=1}^{M} \alpha_i b_i(\mathbf{u})
\]

where \( \alpha_i \geq 0, i = 1, \ldots, M \) are the mixture weights, with \( \sum_{i=1}^{M} \alpha_i = 1 \), and \( b_i(\mathbf{u}) \), \( i = 1, \ldots, M \) are the \( K \)-variate Gaussian densities with mean vector \( \mu_i \) and covariance matrix \( \Sigma_i \).

GMMs can assume several different forms, depending on the type of covariance matrices. The two most widely used are full and diagonal covariance matrices. The number of GMM parameters that have to be estimated from sample or training data is given by \( M(K^2 + 3K + 2) \) for full matrices and \( M(2K + 1) \) for diagonal matrices.

The parameters of the GMM are commonly estimated via the EM algorithm [8]. The algorithm iterations produce a sequence of models with monotonically nondecreasing (log) likelihood values. Though the EM algorithm converges to a maximum likelihood it has a few drawbacks: it is a greedy algorithm and may converge to a local maximum and not the global maximum. GMMs produced by the EM algorithm are, consequently, sensitive to initialization. We use the \( k \)-means algorithm to initialize the GMM parameters.

2.2 MMSE Estimation Using GMMs
The goal in GMM-based MMSE estimation is to find a mapping or regression function, \( \hat{f}(x) \), that minimizes the mean squared error, \( \epsilon_{\text{MSE}} \), between predictor variables (\( x \)) and the target variable (\( y \)), where

\[
\epsilon_{\text{MSE}} = E[(y - \hat{f}(x))^2].
\]

It is known that the mean squared error (2) is minimized when \( \hat{f}(x) = E[y|x] \), the conditional expectation of the target variable, given the predictor vector.

GMM-based estimators rely on modelling the joint density of the \( K \)-dimensional predictor variables with the target
variable using (1) with \( u = [y, x]^T \). The covariance matrix of the \( i^{th} \) GMM component becomes

\[
\Sigma_i = \begin{pmatrix} \Sigma^{yy}_i & \Sigma^{yx}_i \\ \Sigma^{xy}_i & \Sigma^{xx}_i \end{pmatrix}.
\]

Given the GMM parameters, the MMSE regression function is given by [1]

\[
\hat{f}(x) = E[y|x] = \sum_{i=1}^{M} h_i(x) \mu_i^y + \Sigma_i^{yx} (\Sigma_i^{xx})^{-1} (x - \mu_i^x). 
\] (3)

The above GMM estimator or GMM regressor function is a weighted sum of linear models, where the weight \( h_i(x) \) is the probability that the \( i^{th} \) Gaussian component generated the vector \( x \) and given by

\[
h_i(x) = \frac{\alpha_i}{\sum_{k=1}^{M} \alpha_k} e^{-\frac{1}{2} (x - \mu_i^x)^T (\Sigma_i^{xx})^{-1} (x - \mu_i^x)}. \quad (4)
\]

If the covariance matrices are restricted to be diagonal, (3) simplifies to

\[
\hat{f}(x) = \sum_{i=1}^{M} h_i(x) \mu_i^y. \quad (5)
\]

3. FEATURE SELECTION

The primary objective in the feature selection problem is to find, amongst \( n \) candidate features, a subset of variables \( \{x_1, \ldots, x_m\}, m \leq n \), and a mapping \( \hat{f}(x_1, \ldots, x_m) \), such that \( \hat{f} \) yields a good estimate of the response variable \( y \). Presently, CART [5] and MARS [6] have been used as feature selection algorithms and/or regressors in many diverse problems, including speech quality prediction [9]. They are used here to benchmark our proposed feature selection algorithm, which is described below.

3.1 Proposed Method

It is argued in [11] that the GMM estimator has interesting relations to models such as CART and MARS in the sense that the mixture of Gaussians competitively partitions the feature space and learns a linear regression surface on each partition. Thus, it seems evident that one should use the GMM estimator to sift out the most relevant variables. Here we propose a sequential feature selection algorithm that progressively constructs \( \hat{f} \) using (3) or (5).

The proposed algorithm starts with an empty feature set and features from a candidate feature set are added to the set progressively. To determine which candidate feature to add, the algorithm tentatively adds to the current feature set one feature that is not already selected to form an augmented feature set. The joint density of the target variable and the augmented feature set is modelled with a GMM, with model parameters \( \lambda = (\alpha, \mu, \Sigma) \) estimated using the EM algorithm. The accuracy of the GMM estimator using \( \lambda \) is then calculated. The above is repeated for every candidate feature and corresponding GMM. The candidate feature that produces the least regression error is admitted into the current feature set to form an updated feature set. The algorithm stops when the desired number of features has been selected.

It is worth mentioning that for each candidate feature the best number of Gaussian components, \( M \), in (1) can be determined by checking different values of \( M \). With a corresponding increase in computational complexity, multiple features can also be tested and selected per iteration. Using the notation “EM” to stand for GMM parameter estimation via the EM algorithm, \( \hat{f}_k \) for the regression function with \( k \) variables, and \( D \) for the desired number of features, the algorithm can be summarized as follows:

\[
\begin{align*}
\text{Initialization:} & \quad I = \{1, \ldots, n\}, S = \emptyset, k = 1; \\
\text{Step 1:} & \quad \lambda_0 \leftarrow \text{EM}(y, S \cup \{x_i\}), \forall i \in I; \\
\text{Step 2:} & \quad i_k = \arg \min_{i \in I} \sum_{j \in S} (y_j - \hat{f}_k(S \cup \{x_i\})/2; \\
\text{Step 3:} & \quad I \leftarrow I - \{i_k\}, S \leftarrow S \cup \{x_i\}, k \leftarrow k + 1.
\end{align*}
\]

Go to step 1 if \( k < D \), else stop.

Also note that for full covariance matrices the number of parameters that need to be estimated scales quadratically with the feature space dimension. When dealing with limited data, as in our case, severe problems arise due to singularities and local maxima in the log-likelihood function. Here we avert ill-conditioning by adding a small diagonal matrix, namely \( \varepsilon I_{nxn} \), to each covariance matrix in each M-step iteration of the EM algorithm. Typically, the optimal value for \( \varepsilon \) is not known \textit{a priori}. A simple procedure used here is to vary \( \varepsilon \) over a range of values and choose the value that leads to the best performance on the validation data set. We varied \( \varepsilon \) from \( 10^{-2} \) to \( 10^{-5} \) and the value that led to best performance was \( \varepsilon = 10^{-9} \).

We dedicate the next section to testing the proposed feature selection algorithm on a GMM-based speech quality estimation task. We compare with GMM estimators trained on features selected by CART or MARS. We present results for both diagonal and full covariance GMMs.

4. EXPERIMENT SETUP

The GMM for speech quality estimation is built on perceptual feature variables obtained by classifying perceptual distortions under a variety of contexts to form a pool of 206 candidate features. In [4], the top-5 most important feature variables as ranked by CART and MARS are used for training GMM estimators. The target variable is the subjective quality rating represented by the Mean Opinion Score (MOS) [10], which falls between 1 and 5.

We compare GMM estimators trained on features selected by our proposed feature selection algorithm to estimators trained on features selected by CART or MARS. Thirteen MOS labelled speech databases are used, containing a total of 5864 speech files. We use 10-fold cross validation to provide some robustness in the performance evaluation. Estimation performance is assessed by the correlation (\( R \)) between subjective MOS \( w_i \) and estimated MOS \( \hat{y}_i \), using Pearson’s formula

\[
R = \frac{\sum_{i=1}^{N} (w_i - \bar{w})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (w_i - \bar{w})^2 \sum_{i=1}^{N} (y_i - \bar{y})^2}}.
\]
where \( \bar{w} \) is the average of \( w_i \), and \( \bar{y} \) is the average of \( y_i \). MOS measurement accuracy is assessed using the root-mean-square MOS error (RMSE)

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (w_i - y_i)^2}{N}}.
\]

For this experiment we check all permissible values of \( M \) at each iteration. To allow comparisons with [4] we choose the top-5 features and restrict \( M \leq 5 \) in order to maintain an adequate training ratio (ratio between the number of parameters that have to be estimated during the training phase and the total number of files in the training set) of 37 for full covariance matrices and 81 for diagonal matrices.

Let \( M_i \) be the number of Gaussian components chosen in iteration \( i \) of the proposed algorithm, it was found that the following combinations were often selected throughout the ten cross validation trials:

- Diagonal: \( M_1 = 4, M_2 = M_3 = M_4 = M_5 = 5 \);
- Full: \( M_1 = 2, M_2 = 3, M_3 = M_4 = 4, M_5 = 5 \).

Note that for the five algorithm iterations (\( D=5 \)) used in this experiment the number of Gaussian components either increases or stays the same as the algorithm progresses. As expected, full covariance GMMs use fewer Gaussian components at the beginning, and the number of components increases with the number of features.

Tables 1 and 2 compare performance figures for a five-component GMM estimator designed using the proposed algorithm to that of an estimator designed using CART or MARS, for diagonal and full covariance matrices, respectively. The column “\( \% \uparrow R \)” indicates percentage improvement in \( R \) found by using features selected with our proposed method. The percentage improvement is given by

\[
% \uparrow R = \frac{R_{\text{new}} - R_{\text{old}}}{1 - R_{\text{old}}} \times 100\%
\]

where \( R_{\text{new}} \) and \( R_{\text{old}} \) are the correlation obtained using the proposed method and using CART or MARS, respectively. The improvement indicates percentage reduction of the gap to perfect correlation. In turn, column “\( \% \downarrow \)” indicates percentage reduction in \( \text{RMSE} \).

As can be seen, our proposed algorithm outperforms both benchmark algorithms. For diagonal GMM estimators we see an average improvement in \( R \) of 26.95\% and 38.94\%, and an average decrease in \( \text{RMSE} \) of 13.93\% and 24.16\% when compared to CART and MARS, respectively. An average improvement in \( R \) of 31.10\% and 20.01\%, and an average decrease in \( \text{RMSE} \) of 19.07\% and 11.96\% is achieved for full GMM estimators. Also note that an average improvement over PESQ [11], the current “state-of-art” speech quality prediction algorithm, of 26.11\% in \( R \) and 18.04\% in \( \text{RMSE} \) is attained with our full GMM estimators.

One of the drawbacks of using CART and MARS for speech quality estimation was highlighted in [4] and consisted in the fact that features selected by the data mining algorithms had significant correlation amongst them. Diagonal covariance GMM estimators, consequently, presented only modest performance figures and this was attributed to the fact that the use of a small number of diagonal Gaussian components (\( M = 5 \)) was insufficient to model or compensate for the correlation between features. By looking at Tables 1 and 2 we see that CART selected features outperform MARS selected features with diagonal GMM estimators. This does not hold true when using full GMM estimators, suggesting that MARS selected features are more correlated.

Furthermore, in [4] a prominent vertical alignment of points in the subjective MOS versus estimated MOS scattered plots (vide Fig.1(a)) suggested that full covariance GMM estimators were needed in order to predict the residual variation in subjective MOS. If one insists on using diagonal GMM, the problem is mitigated by using the proposed feature selection algorithm, as shown in Fig.1(b). Note that the vertical alignment of points is considerably less accentuated than CART, reflecting the performance improvements shown in Table 1. For MARS, the scattered plot is similar to CART and is omitted for brevity.

Observe in Figure 1(a) the five discrete estimated MOS values associated with the five diagonal Gaussian components (see (5)) are prominently indicated by the horizontal locations of the vertical clusters. In this case, the weights in (5) serve the sole purpose of switching between the five discrete values.

5. CONCLUSION

A feature selection algorithm is proposed for estimation based on Gaussian mixture models. The algorithm is targeted to applications that make use of GMM estimators, as features are selected to minimize squared GMM estimation errors. An experiment consisting of predicting the subjective quality rating of speech is performed. Simulation results show that
Table 1: Performance Comparison: diagonal covariance matrices

<table>
<thead>
<tr>
<th>Cross Validation</th>
<th>Proposed</th>
<th>CART</th>
<th>MARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trials</td>
<td>R</td>
<td>RMSE</td>
<td>%↑</td>
</tr>
<tr>
<td>Trial 1</td>
<td>0.8578</td>
<td>0.4390</td>
<td></td>
</tr>
<tr>
<td>Trial 2</td>
<td>0.8539</td>
<td>0.4623</td>
<td></td>
</tr>
<tr>
<td>Trial 3</td>
<td>0.8530</td>
<td>0.4479</td>
<td></td>
</tr>
<tr>
<td>Trial 4</td>
<td>0.8732</td>
<td>0.4448</td>
<td></td>
</tr>
<tr>
<td>Trial 5</td>
<td>0.8416</td>
<td>0.4585</td>
<td></td>
</tr>
<tr>
<td>Trial 6</td>
<td>0.8694</td>
<td>0.4266</td>
<td></td>
</tr>
<tr>
<td>Trial 7</td>
<td>0.8740</td>
<td>0.4305</td>
<td></td>
</tr>
<tr>
<td>Trial 8</td>
<td>0.8656</td>
<td>0.4409</td>
<td></td>
</tr>
<tr>
<td>Trial 9</td>
<td>0.8721</td>
<td>0.4623</td>
<td></td>
</tr>
<tr>
<td>Trial 10</td>
<td>0.8677</td>
<td>0.4341</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Performance Comparison: full covariance matrices

<table>
<thead>
<tr>
<th>Cross Validation</th>
<th>Proposed</th>
<th>CART</th>
<th>MARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trials</td>
<td>R</td>
<td>RMSE</td>
<td>%↑</td>
</tr>
<tr>
<td>Trial 1</td>
<td>0.8931</td>
<td>0.3830</td>
<td></td>
</tr>
<tr>
<td>Trial 2</td>
<td>0.8917</td>
<td>0.4005</td>
<td></td>
</tr>
<tr>
<td>Trial 3</td>
<td>0.8835</td>
<td>0.3930</td>
<td></td>
</tr>
<tr>
<td>Trial 4</td>
<td>0.9023</td>
<td>0.3907</td>
<td></td>
</tr>
<tr>
<td>Trial 5</td>
<td>0.8852</td>
<td>0.3923</td>
<td></td>
</tr>
<tr>
<td>Trial 6</td>
<td>0.8919</td>
<td>0.3889</td>
<td></td>
</tr>
<tr>
<td>Trial 7</td>
<td>0.8953</td>
<td>0.3955</td>
<td></td>
</tr>
<tr>
<td>Trial 8</td>
<td>0.9075</td>
<td>0.3644</td>
<td></td>
</tr>
<tr>
<td>Trial 9</td>
<td>0.8963</td>
<td>0.3889</td>
<td></td>
</tr>
<tr>
<td>Trial 10</td>
<td>0.9047</td>
<td>0.3708</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GMM estimators designed using our proposed algorithm outperform two benchmark algorithms. Furthermore, we have also shown that features selected by the proposed algorithm are suitable for diagonal GMM estimators, which incur lower computational complexity.

REFERENCES


