GRADIENT BOOSTING LEARNING OF HIDDEN MARKOV MODELS

Rusheng Hu, Xiaolong Li, and Yunxin Zhao

Department of Computer Science
University of Missouri, Columbia, MO 65211, USA
{rhe02,xlm7b}@mizzou.edu, zhaoy@missouri.edu

ABSTRACT

In this paper, we present a new training algorithm, *gradient boosting learning*, for Gaussian mixture density (GMD) based acoustic models. This algorithm is based on a function approximation scheme from the perspective of optimization in function space rather than parameter space, i.e., stage-wise additive expansions of GMDs are used to search for optimal models instead of gradient descent optimization of model parameters. In the proposed approach, GMD starts from a single Gaussian and is built up by sequentially adding new components. Each new component is globally selected to produce optimal gain in the objective function. MLE and MMI are unified under the *H*-criterion, which is optimized by the extended BW (EBW) algorithm. A *partial extended* EM algorithm is developed for stage-wise optimization of new components. Experimental results on WSJ task demonstrate that the new algorithm leads to improved model quality and recognition performance.

1. INTRODUCTION

Maximum likelihood estimation (MLE) is commonly used for GMD based acoustic models where the estimators can be obtained by the expectation maximization (EM) algorithm. MLE assumes that the training data is distributed as described by the model and the size of training samples is large. However, in speech recognition applications, neither of these assumptions holds. Previous works have shown that discriminative training schemes such as maximum mutual information (MMI) [1] and minimum classification error (MCE) [2] can provide better performance than MLE, where information of both correct and competing classes are used to minimize recognition error rate. It was also shown that further improvements can be made by using the *H*-criterion [3], which is a generalized form of ML and MMI, or *I*-smoothing [1].

Two important issues in EM or extended EM algorithm are local optima and model complexity, which depend on the initialization of mixture components. These problems are even more acute in discriminative training when MLE models are used for initialization, because the discriminative objective function is a different criterion. In MLE, randomized model initialization or model selection methods are often used as solutions to these problems. But for discriminative training, these methods can not be directly applied and MLE is most commonly used for model initialization. Normandin [4] proposed an optimal splitting algorithm for GMDs with MMI criterion. But this algorithm was developed in a model complexity perspective and local optima were not of concern. Chou and Li [5] proposed a method of integrating AdaBoost in MCE training for text classifiers, where AdaBoost is used to find meaningful initializers beyond local optima. Although AdaBoost has been reported in boosting HMM speech recognizers [6], its use for initialization of discriminative training is not straightforward.

The key problem associated with local optima and model complexity is the lack of schemes which can jointly optimize model structure and parameters. In this paper, we present the general framework of gradient boosting learning to address above problems. The theory of gradient boosting learning was first introduced in statistics literature. Friedman developed a general gradient-descent boosting paradigm for additive expansions of functions based on any fitting criterion [7]. This paradigm is extended to estimation of GMD based HMMs in our algorithm where GMDs are additive in nature. In addition, a partial extended EM algorithm for optimal component search is developed based on the *H*-criterion. In this new framework, GMDs are recursively constructed in a greedy manner— an optimal new component is located and inserted to the mixture model. In comparison with conventional algorithms, it offers a mechanism of dynamically allocating new components outside the local optimum regions. Conceptually, this algorithm differs from optimal splitting algorithm in that it uses an optimal insertion step instead of splitting, where the new component is found by a global search to avoid local optima. Note that the optimal splitting in [4] does not guarantee global optimum.

The organization of the rest of the paper is as follows. In section 2 we give general background to gradient boosting learning and its application to ML estimation of GMD based HMMs. In section 3, we apply gradient boosting to HMMs based on the *H*-criterion. In section 4 some key implementation issues are discussed. Experimental results are given in section 5, and conclusions are made in section 6.

2. GRADIENT BOOSTING LEARNING

In conventional parametric methods for estimation of function $F(x;\Lambda)$, model parameters Λ are estimated by optimizing some specified objective function $L(F(x;\Lambda))$. For most $F(x;\Lambda)$ and

This work is supported in part by National Institutes of Health under the grant NIH 1 R01 DC04340-01A2 and the National Science Foundation under the grant NSF EIA 9911095.

$L(F(x;\Lambda))$, closed form solution is difficult to find and numerical optimization methods are used. When steepest-decent method is used, the solution can be expressed as a sum of subsequent steps starting from an initial guess λ_0, i.e., $\Lambda_m = \sum_{i=0}^{m} \lambda_i$, where $\lambda_i = \rho_i g_i, i = 1,..,m$ is the incremental step of size ρ_i taken at the direction g_i. In contrast to conventional methods, gradient boosting learning targets the function approximation problem from the perspective of numerical optimization in function space, rather than parameter space. The solutions seeking are "additive" expansions of the form

$$F(x;\Lambda) = \sum_{m=0}^{M} \alpha_m h(x;\theta_m) \quad (1)$$

where $h(x;\theta_m)$ is a basis function characterized by parameters θ_m, which is usually chosen as the best fit of the gradient in the function space at stage *m*, and α_m is the step size. Given *N* training observations $X = (x_1, \ldots x_N)$, the general paradigm of gradient boosting contains the following steps [7]:

Algorithm 1: Gradient Boost

1. Initialize $F_0(x;\Lambda)$.
2. For *m = 1* to *M* do:
3. $g_i = \left[\frac{\partial L(F(x_i;\Lambda))}{\partial F(x_i;\Lambda)}\right]_{F(x)=F_{m-1}(x)}, i = 1,...,N.$
4. Fit basis function $h(x_i;\theta_m)$ to $\{g_i\}$.
5. $\alpha_m = \arg\max_{\alpha} \sum_{i=1}^{N} L(F_{m-1}(x_i;\Lambda) + \alpha h(x_i;\theta_m)).$
6. $F_m(x;\Lambda) = F_{m-1}(x;\Lambda) + \alpha_m h(x;\theta_m).$
7. End For

The analogy of gradient boosting to steepest-descent gives insight to estimation of GMDs in the model space instead of parameter space. Our goal is to estimate a probability density function $f(x;\Lambda)$ which optimizes some specified objective function $L(f(x;\Lambda))$ with the solution in the form of mixture of Gaussians $f(x) = \sum_{i=1}^{k} \alpha_k N_k(x)$ to obtain largest gain of $L(f(x;\Lambda))$ in a steepest-descent manner.

Special properties associated with GMD estimation present difficulties in direct application of gradient boosting. First, the sequential learning equation in line 6 needs to be constrained by being a proper GMD density function. This can be assured by defining the new GMD to be

$$f_m(x;\Lambda) = (1-\alpha_m) f_{m-1}(x;\Lambda) + \alpha_m N(x;\theta_m) \quad 0 < \alpha_m < 1 \quad (2)$$

Second, fitting the steepest-descent direction in line 4 is sensitive to low valued probabilities. For example, in the case of MLE, the gradient of log-likelihood function is $g_i = \frac{\partial \log f(x_i;\Lambda)}{\partial f(x_i;\Lambda)} = \frac{1}{f(x_i;\Lambda)}$. This implies fitting a bell-shaped Gaussian kernel to the reciprocals of current probabilities, which could approach infinity when $f(x_i;\Lambda)$ is small. Third, steepest-descent methods have known problems of local optima. To overcome these problems, we developed an alternative searching procedure to obtain the basis function in line 4. This scheme consists of candidate generation, re-estimation and selection. In our candidate generation design, all candidates are obtained by randomly splitting the existing Gaussian components, which will maintain appropriate coverage of the model space. Each candidate is re-estimated using local data and its contribution to the improvement in the objective function is measured. The one which contributes the most to the objective function is chosen as the new component. Within this scheme, the entire model space is covered by the globally generated candidates, and hence local optima can be alleviated. More details on new component allocation will be discussed in section 4.

Model complexity is one important issue in GMD estimation. The best value for number of components *M* can be determined by model selection methods, such as BIC, cross-validation, etc. By considering the GMD-related issues and incorporating model selection criterion, the gradient boosting algorithm for *S*-class GMDs $\{f_1,...,f_S\}$ is formulated as following:

Algorithm 2: GMD Gradient Boost

1. Initialize $f_{s,0}(x;\Lambda_s) = N(x;\theta_{s,0})$, *s = 1,...,S*, set *m = 1*.
2. For *s = 1* to *S* do:
3. Find a basis Gaussian $N(x;\tilde{\theta}_{s,m})$.
4. $\{\alpha_{s,m}, \theta_{s,m}\} = \arg\max_{\alpha,\theta} \sum_{i=1}^{N} L((1-\alpha) f_{s,m-1}(x_i;\Lambda) + \alpha N(x_i;\theta)),$ use $N(x;\tilde{\theta}_{s,m})$ found in line 3 for initialization.
5. $f_{s,m}(x;\Lambda) = (1-\alpha_{s,m}) f_{s,m-1}(x;\Lambda) + \alpha_{s,m} N(x;\theta_{s,m}).$
6. Update $f_{s,m}$ using EM [optional].
7. End For
8. Set *m = m+1*.
9. If a stopping criterion is met then exit, else go to line 2.

In line 4, as a modification of line 5 in Algorithm 1, the parameters $\alpha_{s,m}$ and $\theta_{s,m}$ are jointly optimized, which is an inherent property of EM algorithms. In this case the new component found in line 3 is used for initialization. Also note that the re-estimation step in line 6 is not in Algorithm 1. The step is added because in GMD estimation, it is often desirable to tune the model parameters after a structural change caused by insertion of a new component.

There is no closed-form solution for the optimization in line 4. However, it can be viewed as a sequential learning of two component models, with the component $f_{m-1}(x;\Lambda)$ fixed. A partial EM algorithm was proposed in [8] for ML estimation of GMDs, which can be easily extended to the ML estimation of HMMs. The update equations for the m^{th} component of GMD at state *s* are given as following:

$$p(s,m \mid x_i) = p(s \mid x_i) \frac{\alpha_{s,m} N(x_i \mid \mu_{s,m}, \Sigma_{s,m})}{(1-\alpha_{s,m}) f_{s,m-1}(x_i) + \alpha_{s,m} N(x_i \mid \mu_{s,m}, \Sigma_{s,m})} \quad (3)$$

$$\hat{\alpha}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i)}{\sum_{i=1}^{N} p(s \mid x_i)} \quad (4)$$

$$\hat{\mu}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i) x_i}{\sum_{i=1}^{N} p(s,m \mid x_i)} \quad (5)$$

$$\hat{\Sigma}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i)(x_i - \mu_{s,m})(x_i - \mu_{s,m})^T}{\sum_{i=1}^{N} p(s,m \mid x_i)} \quad (6)$$

Normally, a global search as required in line 3 is computationally prohibitive. Since only one component needs to be re-estimated in each iteration, partial EM requires much less computation than full EM. The computational efficiency demonstrated by partial EM is critical in developing a global searching heuristic [8].

3. GRADIENT BOOSTING FOR *H*-CRITERION

Given R training utterances $\{X_1, X_2, \ldots, X_R\}$ with corresponding transcriptions w_r, each consisting of N_r feature vectors, $r = 1,\ldots,R$, the MMI discriminative training criteria is defined by the following expression:

$$L_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(X_r \mid w_r) p(w_r)}{\sum_{w \in M_r} p_\lambda(X_r \mid w) p(w)} \quad (7)$$

where M_r denotes the set of word sequences considered for discrimination in utterance r. The MMI objective function can be generalized to the *H*-criterion [3]:

$$L_H(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(X_r \mid w_r) p(w_r)}{\left(\sum_{w \in M_r} p_\lambda(X_r \mid w) p(w) \right)^h}. \quad (8)$$

As can be seen that both MLE and MMI are special cases of *H*–criterion corresponding to $h = 0$ and $h = 1$ respectively. Maximizing (8) leads to the extended BW algorithm which uses following update equations for mean and variance in each particular dimension of the m^{th} Gaussian component for state s (assuming diagonal covariance matrices) [9]:

$$\hat{\mu}_{s,m} = \frac{\{\theta_{s,m}^{num}(X) - h\theta_{s,m}^{den}(X)\} + D\mu_{s,m}}{\{\gamma_{s,m}^{num} - h\gamma_{s,m}^{den}\} + D} \quad (9)$$

$$\hat{\sigma}_{s,m}^2 = \frac{\{\theta_{s,m}^{num}(X^2) - h\theta_{s,m}^{den}(X^2)\} + D\sigma_{s,m}^2}{\{\gamma_{s,m}^{num} - h\gamma_{s,m}^{den}\} + D}. \quad (10)$$

The extended partial BW update equations are similar to those used in partial EM (equations (5) and (6)) and are omitted here.

A key issue in using update equations (9) and (10) as well as their counterparts in extended partial BW is the selection of proper D and h. Larger D means slow convergence but small D may result in negative variance. If h is large then more discrimination is considered; when h is small, more confusion is allowed. In [10], D was set at 0 and superior performance was observed over conventional MMI by choosing appropriate values of h.

4. APPROXIMATE GRADIENT BOOSTING FOR HMM

There are two reasons that make gradient boosting to discriminative training complex. First, searching for the new component requires evaluating the candidates using entire set of observation data, including correct and competing ones. Even in small vocabulary task, gathering of the required statistics is computationally expensive. Second, step-wise convergence constant D and regularizing factor h need to be determined to achieve fast convergence and prevent over-training.

In order to reduce computation complexity, *1*-best approximation is used in the denominator of the *H*-criterion. The computation of sufficient statistics in each state is further reduced by Viterbi approximation. Our gradient boosting discriminative training algorithm consists of the following steps:

1. Train single Gaussian HMMs and perform recognition on the training set. Obtain the correct set A_s and confusion set B_s for each tied state s using Viterbi segmentation.
2. For each individual state, iteratively insert one optimal component which provides the largest increase in the *H*-criterion. Halt insertion if stopping criterion is met.
3. Re-estimate HMMs after GMDs in all tied states have been filled.

In step 1, Viterbi approximation is used for state time-alignment. The sets A_s and B_s contain indices of observations that are labeled and recognized as state s respectively. The *H*-criterion from section 3 becomes

$$F_s^*(\lambda) = \sum_{r \in A_s} \log p_\lambda(X_r \mid S) - h \sum_{r \in B_s} \log p_\lambda(X_r \mid S) \quad (11)$$

Note that this is in the same form as the objective function in [10], but the re-estimation formulas are modified to be as following:

$$\hat{\mu}_{s,m} = \frac{\sum_{r \in A_s} p(m \mid x_r, s) x_r - h_s \sum_{r \in B_s} p(m \mid x_r, s) x_r + D\mu_{s,m}}{\sum_{r \in A_s} p(m \mid x_r, s) - h_s \sum_{r \in B_{sv}} p(m \mid x_r, s) + D} \quad (12)$$

$$\hat{\sigma}_{s,m}^2 = \frac{\sum_{r \in A_s} p(m \mid x_r, s)(x_r - \mu_{s,m})^2 - h_s \sum_{r \in B_s} p(m \mid x_r, s)(x_r - \mu_{s,m})^2 + D\sigma_{s,m}^2}{\sum_{r \in A_s} p(m \mid x_r, s) - h_s \sum_{r \in B_s} p(m \mid x_r, s) + D} \quad (13)$$

The crucial part in step 2 is allocating optimal new component for insertion into mixture. This is achieved by optimal selection among a set of pre-generated candidates. For each phone state, in order to generate candidates for the $(k+1)^{th}$ component, the training data set is quantized into k disjoint sets: $Q_i = \left\{ x \in X : i = \arg\max_{j=1,\ldots,k} P(j \mid x) \right\}$. Then for each set Q_i, a pair of candidates is generated by randomly splitting Q_i into two disjoint subsets. The means and variances of data sample in these two sets are chosen as candidate parameters, and the initial weight for each candidate component is set to be half the weight of $N(\bullet \mid \theta_i)$. If more candidates are needed from this component, then the random splitting process is carried out repeatedly to obtain the required number of candidates. Assuming m candidates are generated from each existing component, then km candidates are generated for the new component. Each candidate is re-estimated by extended partial EM. In order to keep computation cost low as well as to utilize the strength of exploring local discriminative pattern, the maximum approximation is applied in re-estimation and evaluation of candidates, i.e., only those correct and confusion data which are quantized to Q_i are used for generating candidates from Q_i.

Another important issue is the choice of the parameters h and D in the re-estimation equations. Appropriate values need to be chosen to avoid negative Gaussian variance and slow convergence. The strategies for adjusting the value of D have been widely explored. In [1], it was found that a per-Gaussian level D resulted in improved performance. Since gradient boosting is a stepwise greedy learning of discriminative models, it is natural to employ a per-Gaussian level heuristic to adjust both parameters. To be concise, we only list our heuristics for convergence control through h, which consists of the following steps:

1. For state s, initialize h_s by $h_s = \beta \min(h_{s,\max}, 1 - Err)$, where $h_{s,max}$ is the maximum constant that guarantees a variance floor in state s, Err is the estimator of the state error rate in recognition, and $0<\beta<1$ is a shrinkage variable.
2. Run greedy search for new component. If no valid component is found, set $h_s = \beta h_s$ and continue with a new search.
3. Stop if the increase in the value of the objective function (11) is smaller than a given threshold, or if a pre-defined number of consecutive failures of component search is reached.

5. EXPERIMENTS

The Gradient Boosting (GB) algorithm was evaluated on the WSJ 20K Nov 92 task. The standard training data set (WSJ0+WSJ1) including speech of 284 speakers were used. Speech feature analysis was made at a 10msec frame rate with a 25msec window-size. Speech feature components included 13 MFCCs and their first and second derivatives. Cepstral means were removed for every utterance. The baseline acoustic model was trained using HTK with a fixed number of Gaussians in each mixture.

The GB based acoustic models were trained as the following. First, single Gaussian models were trained using conventional EM and were tied by phonetic decision trees with HTK [11]. Second, in order to generate correct training set As and confusion set Bs for each phone states, Viterbi forced alignment and recognition using the trained single Gaussian models were performed to segment training data into corresponding phone states. Third, GB models were trained for each tied state using segmented data sets, As and Bs, where the maximum allowed number of Gaussians for each phone state was 32. As the last step, an ordinary embedded EM was applied to all the transition probabilities of HMMs for a final optimization.

For the WSJ task, standard trigram language model provided by LDC was used, including 19,982 unigrams, 3,518,595 bigrams, and 3,153,527 trigrams. Only within-word triphone acoustic model was tested, even though GB is equally applicable to cross-word triphone model. One-pass time-synchronous beam search was used for decoding speech with conservative pruning thresholds optimized for testing.

GB models were trained under the H-criterion and the parameters h and D were tuned at a per-Gaussian level. Experimental results from 333 sentences of the si_et_20 evaluation set are listed in Table 1. Word accuracy achieved under the same number of mixture components per mixture density was compared for baseline (MLE) and GB derived models (for GB models, this number is an average over all states). Over the range of studied model complexity, GB models consistently gave lower word error rate than EM trained models, confirming the superior performance of GB training over EM.

Mix. size	10	12	15	16	17
Baseline	88.66	88.59	88.84	89.31	89.33
GB	88.92	89.03	89.51	89.58	89.70

Table 1. Word accuracy of conventional EM and GB

6. CONCLUSION

In this paper, we proposed a new algorithm, Gradient Boosting, for discriminative acoustic model training. We described an effective greedy function learning algorithm under the H-criterion, and extended partial EM serves as an efficient way to explore local discriminative patterns and overcome local optima without much extra computation cost. Experiments conducted on WSJ 20K Nov 92 task showed that the proposed algorithm consistently outperformed the conventional EM. In future study, we will address two implementation related issues: one is to minimize the use of approximations made in section 4, the other is to improve the efficiency in identifying local confusions for new component allocation.

REFERENCES

[1] D. Povey and P.C. Woodland, "Improved discriminative training techniques for large vocabulary continuous speech recognition", *Proc. ICASSP01,* vol. 1, pp. 45-48, 2001.

[2] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition", *IEEE Trans. Speech Audio Proc.,* vol. 5, May 1997.

[3] P.S. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo, and M.A. Picheny, "Decoder selection based on cross-entropies", *Proc. ICASSP88*, pp. 20-23, 1988

[4] Y. Normandin, "Optimal splitting of HMM Gaussian mixture components with MMIE training", *Proc. ICASSP95*, vol. 1, May 1995.

[5] W. Chou and L. Li, "A minimum classification error (MCE) framework for generalized linear classifier in machine learning for text categorization/retrieval", *Proc. ICMLA04,* Dec. 2004.

[6] R. Zhang and A. I. Rudnicky, "Improving the performance of an LVCSR system through ensembles of acoustic models", *Proc. ICASSP03,* April 2003.

[7] J.H. Friedman, "Greedy function approximation: a gradient boosting machine", *Annals of Statist.* 29, pp. 1180, 2001.

[8] J.J. Verbeek, N. Vlassis "Efficient greedy learning of Gaussian mixture models," *Neural Comp.* 5(2): 469-485, 2003.

[9] J. Zheng, J. Butzberger, H. Franco and A. Stolcke, "Improved maximum mutual information estimation training of continuous density HMMs", *Proc. Eurospeech,* vol. 2, pp. 679-682, 2001.

[10] A. Ben-Yishai and D. Burshtein, "A discriminative training algorithm for hidden Markov models", *IEEE Trans. Speech Audio Proc.,* vol. 12, May 2004.

[11] The HTK toolkit, *http://htk.eng.cam.ac.uk/*