COMBINING MAXIMUM LIKELIHOOD AND MAXIMUM A POSTERIORI ESTIMATION FOR DETAILED ACOUSTIC MODELING OF CONTEXT DEPENDENCY

Michiel Bacchiani

AT&T Labs - Research
180 Park Ave., Florham Park, NJ 07932, USA
michiel@research.att.com

ABSTRACT

An algorithm is proposed to build large, highly detailed acoustic models for context dependent units using a limited amount of training data. Robustness of the parameter estimates in face of data sparsity is addressed by using MAP distribution smoothing. Context dependent distributions are first clustered using a decision tree-based algorithm with an ML objective. These decision trees are then extended using a MAP objective. Experimental results show an absolute reduction in the word error rate of 0.7% by extending an existing state of the art ML trained context dependent model.

1. INTRODUCTION

Most state of the art speech recognition systems use Hidden Markov Models (HMM) to model phonetic sub-word units. Since the acoustic realization of these sub-word units are dependent on the neighboring units, accurate acoustic models can be constructed by explicitly modeling the acoustic context in terms of the identities of the neighboring units. More specifically, most systems model *triphones* where the acoustic context is defined in terms of the identity of the phonetic units directly proceeding and following. Modeling context explicitly also introduces a data sparsity problem. Even if a moderate sized phonetic unit inventory (say 50 units) is used, the requirement to have sufficient examples of every possible triphone for density estimation becomes prohibitive. To circumvent this requirement, systems use distribution clustering to define a set of distributions which are shared among all possible context dependent models.

In earlier implementations of distribution clustering, tied-mixture modeling [1] was widely used. In the tied-mixture approach a single codebook of mixture components is trained and distributions are defined as mixtures of Gaussians using distribution specific mixture weights for the mixture components defined in the common codebook. An extension of this approach is the phonetic tied mixture system where phone specific codebooks are used[2]. Further extension leads to the state clustering approach used in most current speech recognition systems. In this approach all tied context dependent models share the same *state clustered* distribution [4] or share the same codebook with distributions defined through weight vectors as in the tied-mixture case[5]. Furthermore, due to their ability to generalize to unseen contexts, state of the art systems use decision tree clustering to define the sets of context dependent models that are tied. In [3] the tied-mixture approach is combined with decision tree clustering. In this work, codebooks are defined at intermediate nodes in the decision tree where relatively large pools of data are available. Leaf distributions are then defined by leaf specific weight vectors across the codebook components of the ancestor node. As the leaf specific data is used to estimate the mixture weights alone and the parameters of the mixture components are estimated on a larger pool of data, the resulting leaf distributions are robust.

In more recent work, the tied-mixture or state clustering ideas were extended with use of linear transformations where codebooks or state clusters at the leafs are obtained through linear transformation of the parameters of a codebook or state cluster higher up in the tree [6, 7]. Since the number of free parameters in a linear transformation is much smaller than in an additional codebook or state cluster, robust estimates can be obtained using little training data.

Another approach to obtain robust parameter estimates of a large number of distributions given a limited amount of training data is to use parameter smoothing based on maximum a posteriori (MAP) estimation. In [8] context dependent distributions are derived by MAP estimation using the context independent distribution as a prior. The context independent distributions are trained on a large amount of training data as compared to the training data available to the observed context dependent models. Using the observations of the context dependent units and smoothing the model estimates with the context independent distribution provides robust estimates of the model parameters.

In the recently introduced soft-tying technique [10] an initial decision tree-based state clustered system is refined by growing larger decision trees. Then, to ensure robust parameter estimates of the leaf distributions, non-reciprocal data sharing is implemented between similar leafs (as de-

fined by a similarity measure between leaf distributions).

In this paper, a similar system refinement approach as that of [10] is proposed. Starting from an initial decision tree-based state clustered system, the decision trees are extended, however in contrast to [10], robustness of the parameter estimates at the leaf nodes is ensured by use of the larger pool of data higher up in the tree. In this respect, the approach is similar to that of tied mixtures. In contrast to both the soft tying and tied-mixture approaches, the proposed algorithm uses MAP estimation to obtain smoothed leaf distributions. Leaf distributions of the extended tree are smoothed against the distribution of the ancestor node that was a leaf node in the initial tree.

2. ALGORITHM

The most commonly used distribution sharing technique uses decision trees. A decision tree is constructed for each context independent phone state. The decision trees define the inventory of shared distributions and define a mapping from all possible context dependent realizations to the shared distributions. Let $d = T_p(C)$ denote that context C for phone state p maps to distribution d. Furthermore, let the set of shared distributions be denoted as $\mathcal{D}$ and the set of all possible contexts as $\mathcal{C}$. The decision tree growing algorithm designs the mappings $T_p(C)$ using the training data likelihood given the Maximum Likelihood (ML) estimated models of $\mathcal{D}$. Although Gaussian mixture densities are used for the state emissions of $\mathcal{D}$, Gaussian distributions are used in the tree design because they admit the use of finite size sufficient statistics and a closed form solution for the likelihood ratios to evaluate the decisions[4].

The decision tree state clustering algorithm first computes sufficient statistics for all the unique context dependent units observed in the training data. Let $\mathcal{S}_p$ with $\mathcal{S}_p \subset \mathcal{C}$ denote the set of observed contexts of state p. The set of sufficient statistics is then partitioned according to phonetic class membership questions about the unit contexts. The partitioning is performed iteratively in a greedy fashion, evaluating the merit of a partitioning step by likelihood ratios of the data modeled by the ML parent and child distributions. After the sharing configuration is defined, the training data is used to estimate Gaussian mixtures for the shared distributions, again using the ML criterion (usually using the Expectation-Maximization (EM) algorithm).

To ensure sufficient training data for each shared distribution in the mixture estimation phase, the decision tree clustering algorithm is constrained to ensure a minimum number of observations at each leaf node. Unless the design intends to trade off system size for accuracy, the leaf occupancy threshold needs to balance modeling detail and generalization of the shared distributions. If the leaf occupancy threshold is too large, the number of shared distributions is too small and the model lacks detail. If the leaf occupancy threshold is too small, the number of shared distributions is too large and likelihood improvements modeling the training data do not provide improvements in modeling of test data (i.e. the models are over-fitting the training data).

Building the ML decision trees and training the Gaussian mixtures for the leafs distributions is the first step in the proposed algorithm. The decision trees derived by this step will be referred to as the *initial* trees and mixture model.

The next step in the algorithm is a statistics collection step. For every observation time $t \in T$, the posterior probability

$$\gamma_{d,m}(t) = P(q_{d,m}(t) \mid \mathcal{O}, \mathbf{\Theta})$$

is computed; the probability of being in mixture component m of distribution d at time t given the observations $\mathcal{O} = \{o(1), o(2), \cdots, o(T)\}$ and model parameters $\mathbf{\Theta}$. Using these posterior probabilities, the statistics

$$\nu_{p,s,m} = \sum_{t=1}^{T} \gamma_{T_p(s),m}(t), \tag{1}$$

$$\phi_{p,s,m} = \frac{\sum_{t=1}^{T} \gamma_{T_p(s),m}(t) o(t)}{\sum_{t=1}^{T} \gamma_{T_p(s),m}(t)} \tag{2}$$

and

$$\rho_{p,s,m} = \frac{\sum_{t=1}^{T} \gamma_{T_p(s),m}(t) \mathrm{diag}(o(t)o(t)^T)}{\sum_{t=1}^{T} \gamma_{T_p(s),m}(t)} \tag{3}$$

are computed for all mixture components m of all phone states p in all contexts s. In other words, the algorithm collects first and second order statistics for every observed *unclustered*, unique context dependent unit using the posterior probabilities based on the tied-mixture distributions. Consider the set of contexts $\mathcal{S}_{p,d} = \{\forall s \mid T_p(s) = d\}$ for one of the leaf distributions d of the initial model then the sums $\sum_{s \in \mathcal{S}_{p,d}} \nu_{p,s,m}$, $\sum_{s \in \mathcal{S}_{p,d}} \phi_{p,s,m}$ and $\sum_{s \in \mathcal{S}_{p,d}} \rho_{p,s,m}$ are the E-step statistics that would be collected to perform an EM training iteration of that distribution.

Then the decision trees are grown further, again partitioning the data on the basis of phonetic set membership questions about the unit contexts. The merit of a question is still evaluated on the basis of likelihood ratio tests as before, however, instead of using ML estimation of the new leaf distributions, MAP estimation is used instead. Consider the leaf distribution d of phone state p. A context question splits the set of contexts $\mathcal{S}_{p,d}$ assigned to that leaf into two subsets $\mathcal{S}^l_{p,d}$ and $\mathcal{S}^r_{p,d}$ with $\mathcal{S}_{p,d} = \mathcal{S}^l_{p,d} \cup \mathcal{S}^r_{p,d}$. The left and right statistics sets are then computed by summing the statistics (1)-(3) for all contexts in the left and right subsets. Let the subset statistics for $q \in \{l, r\}$ be defined as $n^q_m = \sum_{s \in \mathcal{S}^q_{p,d}} \nu_{p,s,m}$, $f^q_m = \sum_{s \in \mathcal{S}^q_{p,d}} \phi_{p,s,m}$, and $r^q_m = \sum_{s \in \mathcal{S}^q_{p,d}} \rho_{p,s,m}$. Furthermore, let the ancestor node of d that was a leaf node in the initial tree be denoted as $a_{p,d}$. The distribution of $a_{p,d}$ is then used as the prior for the MAP estimate of the distribution d and the estimates

for the mixture weights, means and variances are derived following [8, 9] as

$$\hat{\omega}^q_{d,m} = \frac{\omega_{a_{d,m}}\tau_w + n^q_m}{\tau_w + \sum_{c=1}^{M_d} n^q_c}, \quad (4)$$

$$\hat{\mu}^q_{d,m} = \frac{\mu_{a_{d,m}}\tau_w + f^q_m}{\tau_w + \sum_{c=1}^{M_d} n^q_c} \quad (5)$$

and

$$\hat{\sigma}^q_{d,m} = \frac{\sigma_{a_d,m}\tau_v + \varphi\tau_v + \chi}{\tau_v + \sum_{c=1}^{M_d} n^q_c} \quad (6)$$

with

$$\varphi = \text{diag}((\mu_{a_{d,m}} - \hat{\mu}^q_{d,m})(\mu_{a_{d,m}} - \hat{\mu}^q_{d,m})^T)$$

and

$$\chi = r^q_m - 2\text{diag}(f^q_m \hat{\mu}^{qT}_{d,m}) + n^q_m \text{diag}(\hat{\mu}^q_{d,m}\hat{\mu}^{qT}_{d,m})$$

and $\omega_{a_{d,m}}$, $\mu_{a_{d,m}}$ and $\sigma_{a_{d,m}}$ denoting the mixture weight, mean and variance of the ancestor node distribution and M_d denoting the number of mixture components in the distribution. The τ parameters are of the conjugate prior density and will affect how quickly the mode of the posterior distribution moves to the ML estimate with more observations. If the τ parameters are all set to zero, the algorithm reduces to the ML tree building algorithm used in the design of the initial tree. If all the τ parameters are set to $+\infty$ the likelihood ratio of any proposed question will be one and the initial tree will not be extended any further. Note that unlike in the ML tree building algorithm, no leaf observation thresholds need to be imposed as the MAP smoothing will back off to a valid distribution (the prior) if no observations are available.

3. EXPERIMENTAL RESULTS

The algorithm was tested on a Voicemail recognition task as described in [11] although another training and test set partition was used. The data set available for training consists of approximately 98 hours of speech. The corpus was manually transcribed at the word level. The messages were collected from 137 voice mailboxes of AT&T Lab - research employees at our Florham Park site. The training set contained 4375 messages from 1468 male speakers and 4716 messages from 1096 female speakers. As assessed by the transcribers, 8290 messages were from regular handsets, the remaining messages were from other types of telephones such as cellular or speakerphones. Also by assessment of the transcribers, 7993 messages were from native speakers. The training corpus consists of 942272 word tokens from 13990 unique words. The test set consists of 2.5 hours of speech containing 27225 word tokens. The recordings were digitized at a sampling rate of 8kHz and encoded as 8-bit μ-law samples.

An initial dictionary was constructed using the AT&T Labs NextGen Text To Speech system for all unique words observed in the training set. The final dictionary was then produced by hand editing. The dictionary used 42 phonemic sub-word units, 5 noise units and 1 silence unit. A trigram language model containing 131623 bigrams and 267967 was constructed from the training set transcriptions.

Number of leaf distributions ($\times 1000$)	Word Error Rate (%)
9.1	27.9
20	27.5
27	27.3
32	27.3
45	27.5

Table 1. Word error rate of systems with different numbers of leaf distributions using MAP estimation of the mixture weight and mean parameters with $\tau_w = 10$. The error rate of the baseline 8016 distribution ML system is 28.0%

The speech was parameterized using Mel frequency cepstral coefficients. The feature vectors consisted of 12 cepstral coefficients, an energy coefficient and their first and second order derivatives (39 dimensional features). The HMM topologies used were 3 state left-to-right for all phone models except silence which was modeled as a single state HMM. The initial tree was constructed from sufficient statistics for 90906 observed unique phone states in triphone context. The 5 noise models and silence were modeled as context independent units. For the 126 speech states, decision trees were grown with 8016 leaf distributions. The leaf occupancy threshold was set to 1000 frames. The occupancy threshold was chosen empirically to optimize accuracy on a test set. The decision tree growing algorithm represented leaf distributions (and hence the sufficient statistics) as full covariance Gaussians. After the decision trees were grown, 12 component mixture distributions were trained for the tied states as well as the context independent models using the EM algorithm. Details on the used training approach can be found in [11].

The initial model was used to generate lattices for rescoring. The first best transcriptions from this recognition pass had a word error rate of 28.2%. A rescoring pass using this same model resulted in an error rate of 28.0% due to the difference in treatment of the back-off cost of the language model in a first and rescoring pass. All experimental results reported here are rescoring error rates using the lattices produced in the first pass.

The initial tree was then extended and a new mixture model was obtained using the described algorithm. In this experiment only mixture weights and means were updated, the variances of the initial model were retained. The τ_w parameter was fixed at 10. Table 1 shows the error rates of the systems with a varying number of leaf distributions. The performance peaks for the 27k distribution system providing a 0.7% absolute improvement in the word error rate over the baseline system.

τ_w	Word Error Rate (%)
0	27.7
10	27.3
20	27.3
100	27.4

Table 2. Word error rate of systems trained with different τ_w parameters but with a fixed number of leafs (27k). Only mixture weight and mean parameters are obtained by MAP estimation, variances are fixed.

τ_v	Word Error Rate (%)
100	27.4
200	27.3
300	27.4
400	27.4

Table 3. Word error rate of systems trained with different τ_v parameters but with a fixed number of leafs (27k) and the τ_w parameter fixed at 10. Mixture weights, mean and variance parameters are obtained by MAP estimation.

Table 2 shows the word error rate of a 27k distribution system when the τ_w parameter is varied but the number of leaf distributions is kept fixed. In this experiment, the variance is again kept fixed. The results shows that ML estimation of the weights and means ($\tau_w = 0$) provides little gain over the baseline system in contrast to the MAP estimated systems.

Table 3 shows the word error rate of a 27k distribution system with the τ_w parameter fixed at 10 but now including variance estimation. The table shows the performance of the system when varying the τ_v parameter. It can be observed that variance estimation does not provide any additional performance gains.

4. CONCLUSION

The proposed algorithm takes advantage of the MAP training algorithm to allow the estimation of a large acoustic model with limited training data. An absolute improvement of 0.7% in the word error rate on a voicemail transcription tasks over a state of the art ML trained model shows the viability of the proposed algorithm. The experimental results show the most benefit from MAP smoothed estimates of the mixture weights and means and no additional benefit from re-estimation of the variances. ML re-estimation of mixture weight and mean parameters alone does not perform as well as MAP re-estimation of those parameters showing the benefit from smoothing with the prior distribution.

5. REFERENCES

[1] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 38, No. 12, pp. 2033-2045, 1990.

[2] D. B. Paul, "The Lincoln robust continuous speech recognizer," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* Vol. 1, pp. 449-452, 1989.

[3] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries and M. Westphal, "The Karlsruhe-Verbmobil speech recognition engine," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* Vol. 1, pp. 83-86, 1997.

[4] P. C. Woodland, J. J. Odell, V. Valtchev and S. J. Young, "Large vocabulary continuous speech recognition using HTK," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* Vol. 2, pp 125-128, 1994.

[5] V. V. Digalakis, P. Monaco and H. Murveit, "Genones: generalized mixture tying in continuous hidden Markov model-based speech recognizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 4, No. 4, pp. 281-289, 1996.

[6] A. Sankar, "Robust HMM Estimation with Gaussian Merging-Splitting and Tied-Transform HMMs," In *Proceedings of the International Conference on Spoken Language Processing,* 1998.

[7] E. Bocchieri, "Phonetic Context Dependency Modeling by Transform," In *Proceedings of the International Conference on Spoken Language Processing,* Vol. 4, pp. 179-182, 2000.

[8] J-L. Gauvain and C-H. Lee, "Bayesian Learning of Gaussian Mixture Densities for Hidden Markov Models," In *Proceedings of the DARPA Speech and Natural Language Workshop,* pp. 271-277, 1991.

[9] J-L. Gauvain and C-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 2, No 2., pp. 291-298, 1994.

[10] X. Luo and F. Jelinek, "Nonreciprocal Data Sharing in Estimating HMM Parameters," In *Proceedings of the International Conference on Spoken Language Processing,* 1998.

[11] M. Bacchiani, "Automatic transcription of voicemail at AT&T," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* Vol. 1, pp. 25-28, 2001.