

Subspace Distribution Clustering Hidden Markov Model

Enrico Bocchieri, *Member, IEEE*, and Brian Mak, *Member, IEEE*

Abstract— Most contemporary laboratory recognizers require too much memory to run, and are too slow for mass applications. One major cause of the problem is the large parameter space of their acoustic models. In this paper, we propose a new acoustic modeling methodology which we call *subspace distribution clustering hidden Markov modeling* (SDCHMM) with the aim at achieving much more compact acoustic models. The theory of SDCHMM is based on tying continuous density hidden Markov models (CDHMMs) at a new finer sub-phonetic unit, namely the subspace distribution. SDCHMMs can be converted from CDHMMs by projecting the distributions of the CDHMMs onto orthogonal subspaces, and then tying similar subspace distributions over *all* states and *all* acoustic models in each subspace. By exploiting the combinatorial effect of subspace distribution encoding, all original full-space distributions can be represented by combinations of a small number of subspace distribution prototypes. Consequently, there is a great reduction in the number of model parameters, and thus substantial savings in memory and computation. This renders SDCHMM very attractive in the practical implementation of acoustic models. Evaluation on the ATIS (Airline Travel Information System) task shows that in comparison to its parent CDHMM system, a converted SDCHMM system achieves 7- to 18-fold reduction in memory requirement for acoustic models, and runs 30–60% faster without any loss of recognition accuracy.

Keywords— Hidden Markov modelling, subspace distribution, distribution clustering.

I. INTRODUCTION

THE high computational cost of many state-of-the-art automatic speech recognizers is a major impediment to their deployment in mass applications. A significant challenge is to adjust these recognizers so that they may be run on more affordable machines of lower processing power and smaller memory size *without* losing accuracy. Techniques exist to reduce memory requirement alone, for example, by using simpler but less accurate models, or through data compression [1]. There are also techniques to speed up computation alone: for example, by simply exercising more vigorous pruning schemes, by computing state likelihoods only from a small subset of the most relevant state probability density distributions [2], [3], [4], [5], [6], or by fast-match techniques [7]. However, these techniques are usually done at the expense of recognition accuracy; in the case of computation speedup, more memory is usually required. One of the most effective way to achieve all

the three seemingly conflicting goals — faster recognition speed, smaller memory footprint, and high accuracy — is to reduce the number of parameters in the acoustic models. In general, a smaller model parameter space has the following advantages:

- smaller memory requirement
- faster recognition
- requiring fewer training data
- requiring fewer data for speaker or environment adaptation.

In this paper, we propose a more efficient acoustic modeling methodology to arrive at a more compact recognizer.

A. Subspace Distribution Tying

The most common approach to reducing the number of parameters in acoustic models is parameter tying: Similar structures are discovered among the acoustic models, and they are then tied together to share the same value. With the (limited) amount of training data on hand, parameter tying allows more complex acoustic models to be estimated reliably while the number of model parameters will not grow unchecked. In the past, the technique of parameter tying has been applied successfully at various granularities: Phones (context-independent phones [8], generalized biphones/triphones [9]), states (tied-state HMM [10], [11]), observation distributions (tied-mixture/semi-continuous HMM [12], [13], [14]), and feature parameters [15] have all been tied. The technology trend is to tie acoustic models at finer and finer details so as to maintain good resolution among models as much as possible. In this paper, we propose to push the technique to an even finer sub-phonetic unit — subspace distributions — in the context of hidden Markov modeling. Subspace distributions are the projections of the full-space distributions of an HMM in lower dimensional spaces. The hypothesis is that speech sounds are more alike in some acoustic subspaces than in the full acoustic space. We call our novel HMM formulation “*subspace distribution clustering hidden Markov modeling*” (SDCHMM).

Subspace distribution clustering hidden Markov models (SDCHMMs) can be derived from already existing continuous density hidden Markov models (CDHMMs) without requiring any extra training data nor re-training. The distributions of CDHMMs are projected onto orthogonal subspaces (or streams¹), and similar subspace distributions are then tied into a small number of distribution prototypes over *all* states and *all* acoustic models in each subspace. By

¹In this paper, the two terms, “subspace” and “stream” are used interchangeably to mean a feature space of dimension smaller than that of the full feature space.

Enrico Bocchieri is with the AT&T Labs – Research, Florham Park, NJ 07932, USA. E-mail: enrico@research.att.com.

Brian Mak is with the Department of Computer Science, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. This work is finished when he was a PhD candidate of Oregon Graduate Institute of Science and Technology, Portland, Oregon, USA, and a Research Consultant at the AT&T Labs – Research. E-mail: mak@cs.ust.hk.

exploiting the combinatorial effect of subspace distribution encoding, all original full-space distributions can be closely approximated by some combinations of a small number of subspace distribution prototypes. Consequently, there is a great reduction in the number of model parameters, and thus substantial savings in memory and computation. This renders SDCHMM very attractive in practical implementation of acoustic models.

From the perspective of quantization, one may consider SDCHMM as an approximation to the highly accurate CDHMM, achieving great data compression by subspace distribution quantization. From the perspective of hidden Markov modeling, SDCHMM unifies the theory of CDHMM which employs full-space state probability density distributions and the feature-parameter-tying HMM [16], [15] which is generated by scalar quantization of the distributions. SDCHMM combines the accuracy of CDHMM with the compactness of feature-parameter-tying HMM. In this aspect, it is interesting to compare this work with a similar approach called “split vector quantization” [17], [18] that has been successfully applied to high-quality, low-bit rate speech coding for years. In speech coding, it is known that (full) vector quantization (VQ) results in smaller quantization distortion than scalar quantization at any given bit rate [19]. However, to attain the required high quality in practical telecommunication, full VQ suffers from training, memory, and computation problems much like those of our current complex speech recognizers. Split VQ overcomes the complexity problem of full VQ by splitting the speech vectors into sub-vectors of lower dimensions and quantizing the sub-vectors in their subspaces.

The organization of this paper is as follows. In Section II, we present the theory of SDCHMM. Section III describes an implementation method in which SDCHMMs are converted from CDHMMs through a simple Gaussian clustering algorithm. A coherent definition of the streams are also proposed. The conversion method is evaluated in Section IV on the ATIS task. The effect of different numbers of streams and different amounts of tying will be studied and evaluated on three metrics: accuracy, computation time, and memory requirement. In Section V, we compare the SDCHMM with two similar HMM methodologies. Finally, we draw our conclusions in Section VI.

II. SUBSPACE DISTRIBUTION CLUSTERING HIDDEN MARKOV MODEL

A. Theory of SDCHMM

The theory of SDCHMM is derived from that of the continuous density hidden Markov model (CDHMM). Let us first consider a set of CDHMMs (possibly with tied states) in which state-observation distributions are estimated as mixture Gaussian densities with M components and diagonal covariances. Using the following notations (where, as usual, bold-faced quantities represent vectors):

- \mathbf{O} : an observation vector of dimension D
- $P(\mathbf{O})$: state output probability given \mathbf{O}
- c_m : weight of the m -th mixture component

- $\boldsymbol{\mu}_m$: mean vector of the m -th component
- $\boldsymbol{\sigma}_m^2$: variance vector of the m -th component
- $\mathcal{N}(\cdot)$: Gaussian pdf

the state observation probability is given by

$$P_{CDHMM}(\mathbf{O}) = \sum_{m=1}^M c_m \mathcal{N}(\mathbf{O}; \boldsymbol{\mu}_m, \boldsymbol{\sigma}_m^2), \quad \sum_{m=1}^M c_m = 1. \quad (1)$$

The key observation is that a Gaussian with diagonal covariance can be expressed as a product of subspace Gaussians where the subspaces (or streams) are orthogonal and together span the original full feature vector space. Formally, let us denote the full vector space of dimension D by \mathcal{R}^D with an orthonormal basis, which are composed of the column vectors of the $D \times D$ identity matrix. \mathcal{R}^D is decomposed into K orthogonal subspaces \mathcal{R}^{d_k} of dimension d_k , $1 \leq k \leq K$, with the following conditions:

Condition 1:

$$\sum_{k=1}^K d_k = D \quad (2)$$

Condition 2:

$$\mathcal{R}^{d_i} \cap \mathcal{R}^{d_j} = \emptyset, \quad 1 \leq i \neq j \leq K. \quad (3)$$

Condition 3: The basis of each subspace is composed of a subset of the basis vectors of the full vector space.

Each of the original full-space Gaussians is projected onto each of the K subspaces to obtain K subspace Gaussians of dimension d_k , $1 \leq k \leq K$, with diagonal covariances. That is, Equation (1) can be rewritten as

$$P_{CDHMM}(\mathbf{O}) = \sum_{m=1}^M c_m \left(\prod_{k=1}^K \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2) \right) \quad (4)$$

where \mathbf{O}_k , $\boldsymbol{\mu}_{mk}$, and $\boldsymbol{\sigma}_{mk}^2$ are the projection of the observation \mathbf{O} , and mean and variance vectors of the m -th mixture component onto the k -th subspace respectively.

For each stream, we treat the subspace Gaussians as the basic modeling unit, and tie them across *all* states of *all* CDHMM acoustic models. Hence, the state observation probability in Equation (4) is modified as

$$P_{SDCHMM}(\mathbf{O}) = \sum_{m=1}^M c_m \left(\prod_{k=1}^K \mathcal{N}^{tied}(\mathbf{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2) \right). \quad (5)$$

The ensuing HMM will be called the *subspace distribution clustering hidden Markov model* (SDCHMM). Figure 1 shows an extension of various HMM tying schemes to include SDCHMMs. There are 4 streams in the example.

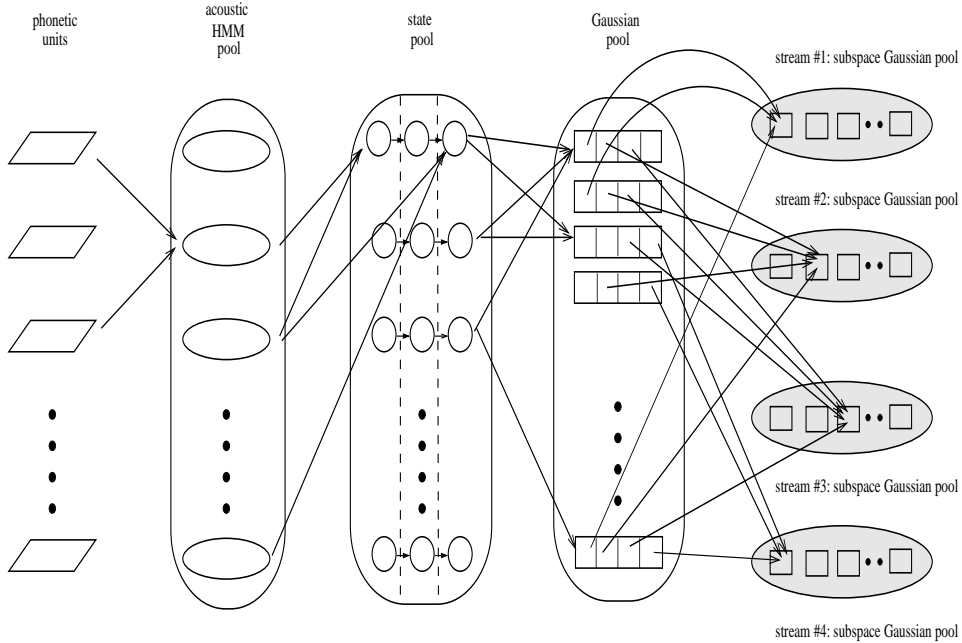


Fig. 1: Subspace distribution clustering hidden Markov models with 4 streams

figure

A.1 Generalization

The foregoing SDCHMM formulation can be generalized to any mixture density insofar as the component pdf $\mathcal{F}(\mathbf{O})$ can be expressed as a product of subspace pdf's of the same functional form. That is,

$$\mathcal{F}(\mathbf{O}) = \prod_{k=1}^K \mathcal{F}(\mathbf{O}_k), \quad (6)$$

provided that the three conditions on the subspaces mentioned above are satisfied. An obvious candidate for this functional is a Gaussian pdf with block-diagonal covariance.

While Gaussians with block-diagonal covariances or other pdf functionals appear intriguing, they have not been widely studied (except, e.g., [20]) in automatic speech recognition. To keep our focus on the main issue of SDCHMM in this paper, we investigate only SDCHMMs based on CDHMMs with mixture Gaussian densities and diagonal covariances.

B. Distribution Clustering

In practice, the proposed SDCHMM as in Equation (5) can be obtained by clustering or quantizing the subspace Gaussians of CDHMMs in each stream. That is, to derive K -stream SDCHMMs from a set of CDHMMs in which there are originally a total of N full-space Gaussian distributions, the subspace Gaussians in each stream are clustered into a small set of L subspace Gaussian prototypes

$$\mathcal{N}^{quantized}(\mathbf{O}_k; \boldsymbol{\mu}_{lk}, \boldsymbol{\sigma}_{lk}^2), \quad 1 \leq l \leq L, \quad 1 \leq k \leq K$$

where $L \ll N$. Each original subspace Gaussian is then “approximated” by its nearest subspace Gaussian prototype

$$\mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2) \approx \mathcal{N}^{quantized}(\mathbf{O}_k; \boldsymbol{\mu}_{lk}, \boldsymbol{\sigma}_{lk}^2)$$

with l being given by

$$l = \underset{1 \leq q \leq L}{\operatorname{argmin}} \operatorname{dist} \left(\mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2), \right.$$

$$\left. \mathcal{N}^{quantized}(\mathbf{O}_k; \boldsymbol{\mu}_{qk}, \boldsymbol{\sigma}_{qk}^2) \right) \quad (7)$$

where $\operatorname{dist}(\cdot)$ measures the distance between two Gaussian distributions.

In this respect, SDCHMMs can be considered as an approximation to the conventional CDHMMs.

C. Why Are SDCHMMs Good?

If the subspace distributions are properly clustered, all original full-space distributions can be represented by some combinations of a small number of subspace distribution prototypes with small quantization errors. The combinatorial effect of subspace distribution encoding can be very powerful: For instance, a 20-stream SDCHMM system with as few as 2 subspace distribution prototypes per stream can represent $2^{20} = 1,048,576$ different full-space distributions. Of course, in reality, more prototypes are required to ensure small quantization errors. This can be achieved with more streams or more prototypes per stream.

SDCHMMs are also computationally efficient because if a small number of the subspace Gaussians are shared by a

large number of full-space Gaussian components, all these subspace Gaussian log likelihoods can be pre-computed once and only once at the beginning of every frame, and their values are stored in lookup tables. During Viterbi decoding [21] of a K -stream SDCHMM system, the log likelihood of a Gaussian component of a state can be computed as the summation of K pre-computed subspace Gaussian log likelihoods and the log mixture weight.

III. MODEL CONVERSION FROM CONTINUOUS DENSITY HMMs

The formulation of the subspace distribution clustering hidden Markov model as of Equation (5) of Section II suggests that SDCHMMs may be implemented in the following two steps as shown in Figure 2:

1. Train continuous density hidden Markov models for all the phonetic units (possibly with tied states), wherein state observation distributions are estimated as mixture Gaussian densities with diagonal covariances.
2. Convert the CDHMMs to SDCHMMs by tying the subspace Gaussians in each stream.

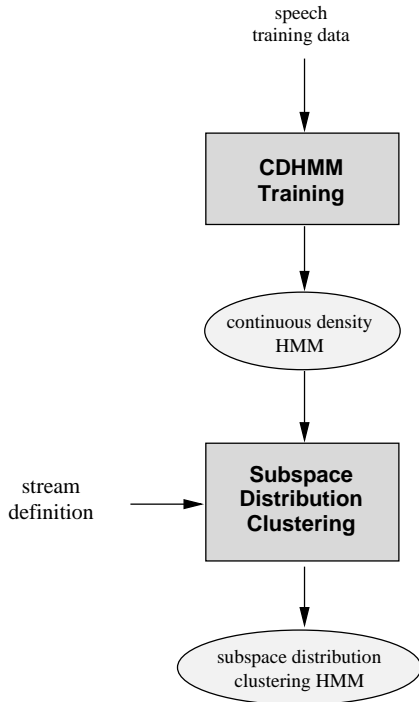


Fig. 2: Conversion of CDHMMs to SDCHMMs figure

Since the training of CDHMMs is well covered in the literature [22], [23], we will not repeat it here. Instead, we assume that a set of (well-trained) CDHMMs is given, and we focus only on the conversion of the CDHMMs to SDCHMMs.

Tying of subspace Gaussians consists of splitting the full speech feature vector space into disjoint subspaces, projecting mixture Gaussians of CDHMMs onto these subspaces, and then clustering the subspace Gaussians into a small

number of Gaussian prototypes in each subspace. In the following, we describe various stream definitions and distribution clustering algorithms to tie subspace Gaussians. They will be evaluated in the next Section.

A. Issue I: Stream Definition

To derive K -stream SDCHMMs, we first have to partition the feature set Ω^D with D features into K disjoint feature subsets Ω^{d_k} with d_k features, $1 \leq k \leq K$. Formally, let \mathcal{P}_K^D be such a partition, then

$$\mathcal{P}_K^D = \left\{ \Omega^{d_k} : \sum_{k=1}^K d_k = D \text{ and } \Omega^{d_k} \cap \Omega^{d_j} = \emptyset \right\} \quad (8)$$

where $1 \leq k \neq j \leq K$.

The partition \mathcal{P}_K^D is optimal if subsequent tying of subspace Gaussians in the feature subspaces of the partition results in minimal total quantization error for a pre-determined number of prototypes and clustering algorithm. In general, the clustering problem cannot be solved analytically, and is tackled numerically using iterative procedures. Since the total number of possible partitions is usually very large, it is not feasible to determine the optimal partition by numerically computing the quantization errors due to all possible candidates. Thus some heuristic approach has to be used to obtain a reasonable partition.

A.1 Common Streams

Our speech input comprises 39 features: 12 MFCCs, normalized power, and their first- and second-order time derivatives. By putting conceptually similar features together in a stream like the commonly-used streams in discrete HMM and semi-continuous HMM, the following “common” definitions of streams are explored:

1-stream definition:

$$12MFCC + 12\Delta MFCC + 12\Delta^2 MFCC + e + \Delta e + \Delta^2 e$$

4-stream definition:

$$\begin{array}{l} 12MFCC \\ 12\Delta MFCC \\ 12\Delta^2 MFCC \\ e + \Delta e + \Delta^2 e \end{array}$$

13-stream definition:

$$12 * \begin{array}{l} MFCC + \Delta MFCC + \Delta^2 MFCC \\ e + \Delta e + \Delta^2 e \end{array}$$

39-stream definition: each 1-dimensional feature is put into one stream.

Note that 1-stream SDCHMMs are identical with the original CDHMMs, and 39-stream SDCHMMs are the same as feature-parameter-tying HMMs.

A.2 Correlated-Feature Streams

We adopt the heuristic that correlated features, by definition, should tend to cluster in a similar manner, and require each stream to have the most correlated features. Intuitively this criterion should result in smaller distortions for the clustered subspace Gaussians. This definition has the additional benefit of providing a single coherent definition for *any* arbitrary number of streams of *any* dimension. Note that, although the features are assumed uncorrelated locally within each Gaussian distribution (with diagonal covariance), during clustering of the subspace Gaussians, it is the global feature correlation that matters.

I. Multiple Correlation Measure

The correlation ρ_{ij} between two variables is commonly measured by Pearson’s moment product correlation coefficient

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}, \quad (9)$$

where σ_i and σ_j are the standard deviations of the i -th and j -th variables respectively, and σ_{ij} is the square root of their covariance. Nevertheless, multiple correlation measures among three or more variables are less studied. In the statistics literature, multiple correlation is usually reduced to a binary correlation [24]. However, this is inappropriate in our context where a multiple correlation measure that emphasizes mutual correlations among all variables at the same time is more desirable. In this paper, we propose a new definition of a multiple correlation coefficient R defined as

$$R \stackrel{\text{def}}{=} 1 - \frac{\text{determinant of correlation matrix of the variables}}{\text{product of variances of the variables}}$$

That is, the multiple correlation coefficient R among k variables is,

$$R = 1 - \frac{\begin{vmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1k} \\ \rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2k} \\ \rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k1} & \rho_{k2} & \rho_{k3} & \cdots & 1 \end{vmatrix}}{\prod_{i=1}^k \sigma_i^2} \quad (10)$$

In the case when there are only two variables, R equals the square of the moment product correlation coefficient.

It can also easily be shown that R has the following desirable properties of a correlation measure:

- $0 \leq R \leq 1$
- when all variables are correlated, i.e. $\forall i, j, \rho_{ij} = 1, R = 1$
- when all variables are uncorrelated, i.e. $\forall i, j, \rho_{ij} = 0, R = 0$.

II. Derivation of Streams

Practically, we apply a greedy algorithm [25] to obtain streams in which the features are most correlated, as depicted in Algorithm 1. It is simple to modify the algorithm in cases when the number of features D is not a multiple of the number of streams K . Since the streams are restricted to have the same dimension, the computation of multiple

TABLE I: ATIS: 20 correlated-feature streams table

STREAM	FEATURES
1	$c_1, \Delta \Delta c_1$
2	$c_2, \Delta \Delta c_2$
3	$c_3, \Delta \Delta c_3$
4	$c_4, \Delta \Delta c_4$
5	$c_5, \Delta \Delta c_5$
6	$c_6, \Delta \Delta c_6$
7	$c_7, \Delta \Delta c_7$
8	$c_8, \Delta \Delta c_8$
9	$c_9, \Delta \Delta c_9$
10	$c_{10}, \Delta \Delta c_{10}$
11	$c_{11}, \Delta \Delta c_{11}$
12	$c_{12}, \Delta \Delta c_{12}$
13	$\Delta c_1, \Delta c_7$
14	$\Delta c_2, \Delta c_6$
15	$\Delta c_3, \Delta c_5$
16	$\Delta c_4, e$
17	$\Delta c_8, \Delta c_9$
18	$\Delta c_{10}, \Delta c_{11}$
19	$\Delta e, \Delta \Delta e$
20	Δc_{12}

correlation coefficients involves only determinants of any $n \times n$ matrices obtained by deleting any $(D - n)$ rows and the corresponding columns from the $D \times D$ feature correlation matrix — which needs to be computed once. As a result, the algorithm is efficient.

Algorithm 1: Selection of the most correlated-feature streams (of the same dimension)

Goal: Given D features, define K n -dimensional streams with $D = nK$.

Step 1. Compute the multiple correlation coefficient among *any* set of n features according to Equation (10). (There are totally $C(D, n)$ coefficients.)

Step 2. Sort the multiple correlation coefficients in descending order, each tagged by an n -feature tuple indicating the features it computes from.

Step 3. Starting from the top, an n -feature tuple is moved from the sorted list to the “solution list” if *none* of its features already appear in any feature tuples of the solution list.

Step 4. Repeat Step 3 until all features appear in the solution list.

Step 5. The feature tuples in the “solution list” are the K -stream definition.

Table I shows the definition of 20 correlated-feature streams generated by Algorithm 1 using 1,000 utterances from the ATIS training corpus. From the definition, $MFCC$ and $\Delta^2 MFCC$ are found mostly correlated.

B. Issue II: Subspace Gaussian Clustering

Two very different clustering schemes are investigated: A bottom-up agglomerative clustering algorithm and a top-down modified k -means (MKM) clustering algorithm.

B.1 Agglomerative Gaussian Clustering Algorithm

The *ensemble merging algorithm* for state tying described in [26] can be applied without modification to cluster subspace Gaussians in each stream instead of HMM states. It is a bottom-up agglomerative clustering scheme in which two subspace Gaussians are merged if they result in minimum increase in distortion (scatter). To avoid an otherwise $O(n^3)$ complexity, the algorithm introduces the heuristic that at each iteration, the Gaussian corresponding to the smallest training ensemble must be merged. As a result, the algorithm has a complexity of $O(n^2)$.

B.2 Modified k -means Gaussian Clustering Algorithm

Algorithm 2 shows a novel $O(JLn)$ modified k -means clustering algorithm which derives L subspace Gaussian prototypes from n Gaussians, in J iterations without using any heuristics. With $JL \ll n$ for large acoustic models, the linearity in n implies improved efficiency (over the ensemble merging algorithm).

Algorithm 2: Modified k -means Gaussians clustering algorithm

Goal: To derive K -stream SDCHMMs with L subspace Gaussian prototypes per stream.

Step 1. Initialization: First train a 1-stream Gaussian mixture model with L components. Project each of the L Gaussian components onto the K subspaces according to the given K -stream specification. The resultant KL subspace Gaussians will be used as initial subspace Gaussian prototypes.

Step 2. Similarly project each Gaussian pdf in the original CDHMMs onto the K subspaces.

Step 3. For each stream, repeat Step 4 & 5 until some convergence criterion is met.

Step 4. Membership: Associate each subspace Gaussian of CDHMMs with its nearest prototype as determined by their Bhattacharyya distance.

Step 5. Update: Merge all subspace Gaussians which share the same nearest prototype to become the new subspace Gaussian prototypes.

To compute the distance between two Gaussians during distribution clustering, we adopt the classification-based Bhattacharyya distance, which is defined as

$$D_{bhat} = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \left[\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right]^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2} \ln \frac{|\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2|}{\sqrt{|\boldsymbol{\Sigma}_1| |\boldsymbol{\Sigma}_2|}} \quad (11)$$

where, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, $i = 1, 2$, are the means and covariances of the two Gaussians [27]. The Bhattacharyya distance has been used in several speech-related tasks [28], [29], [30], leading to good results. The Bhattacharyya distance captures both the first- and the second-order statistics, and is expected to give better clustering results than the Euclidean distortion measure employed in the Agglomerative Gaussian Clustering Algorithm, which makes use of only the first-order statistics.

To initiate the iterative k -means clustering procedure for the conversion of CDHMMs to K -stream SDCHMMs with L subspace Gaussian prototypes per stream, we first train a Gaussian mixture model with L components using 1,000 ATIS training utterances. The L Gaussians are split into L subspace Gaussians for each stream, which are then used as seeds for clustering. If no training data is available, one may, for example, randomly pick L subspace Gaussians from the CDHMMs to start the clustering procedure.

IV. EVALUATION OF SDCHMM

A. The ATIS task

The Air Travel Information System (ATIS) [31] is a medium-vocabulary, spontaneous, and goal-directed speech recognition task. An ATIS system allows users to speak naturally to inquire about air travel information stored as a relational database which is derived from the American Official Airline Guide. To date, the ATIS corpora contain nearly 25,000 utterances with a vocabulary size of 1,536 words. The query database includes information on 23,457 air flights for 46 cities and 52 airports in the United States and Canada. A set of 981 utterances were set aside for the 1994 ARPA-ATIS evaluation.

B. The Baseline CDHMM Recognizer

Our baseline system consists of AT&T's ATIS recognizer used in the 1994 ARPA-ATIS evaluation [32]. The configurations, testing conditions, and performance of both the context-independent (CI) and context-dependent (CD) baseline systems are described in Table II.

The recognizer frontend is based on mel-frequency cepstral analysis of input speech sampled at 16kHz. At every 10ms, 31 mel-frequency energy components are computed from a filter bank by performing an FFT on a *frame* of 20ms of speech. The energies are converted to 12 mel-frequency cepstral coefficients (MFCCs) by cosine transform. *Cepstral mean subtraction* is then performed using the average MFCCs per utterance. Finally a speech feature vector for one frame is composed from 39 components: 12 MFCCs and normalized power, and their first- and second-order time derivatives computed as follows:

$$\begin{aligned} x[t] &= \text{normalized MFCC or power} \\ \Delta x[t] &= 2x[t+2] + x[t+1] - x[t-1] - 2x[t-2] \\ \Delta\Delta x[t] &= \Delta x[t+1] - \Delta x[t-1]. \end{aligned}$$

C. Evaluation

All components of the baseline recognizers are kept intact, except that their acoustic models are converted from

TABLE II: ATIS: Testing conditions and performance of the baseline CI/CD systems

table

CONDITION/PERFORMANCE	CI SYSTEM	CD SYSTEM
#Test Sentences	981 (1994 ARPA-ATIS evaluation set)	
Vocabulary	1,536 words	
Language Model	word-sequence bigram (perplexity ≈ 20)	
#Training Utterances	$\sim 12,000$ ATIS	$\sim 20,000$ ATIS + $\sim 8,000$ WSJ
#HMMs	48	9,769
#States	142	3,916 (tied)
Max. #Mixtures per State	16	20
#Gaussians (39-dimensional)	2,254	76,154
#Acoustic Parameters	178,066	6,016,166
Search	one-pass Viterbi beam search	
Lexical Structure	lexical tree	linear lexicon
Beam-Width	100	170
CPU	150MHZ MIPS R4400	195MHZ MIPS R10000
Word Error Rate	9.4%	5.2%
Time (x real-time)	1.93	7.06
HMM Memory Usage	0.71MB	24MB

CDHMMs to SDCHMMs. The testing conditions are exactly the same as those described in Table II. All subspace Gaussian log-likelihoods are pre-computed at the beginning of each frame, and their values are stored in tables in contiguous memory². In addition, for implementation and system simplicity, all streams are tied to the same number of subspace Gaussian prototypes in all our SDCHMMs.

C.1 I. Stream Definitions and Clustering Algorithms

With the two types of stream definitions of Section III-A and the two clustering algorithms of Section III-B, four different combinations of stream definitions and clustering algorithms are tested using 13 streams:

- common stream definition + ensemble merging
- common stream definition + modified k -means Gaussian clustering
- correlated-feature stream definition + ensemble merging
- correlated-feature stream definition + modified k -means Gaussian clustering.

Thirteen streams are chosen because both the common stream definition and the correlated-feature stream definition readily apply. Each stream consists of exactly three features, and is tied to 8–256 subspace Gaussian prototypes. Each of the ensuing 13-stream SDCHMM systems is then tested on the 1994 ATIS evaluation dataset.

Figure 3(a) and (b) show incremental improvements in recognition performance when correlated-feature streams and/or the modified k -means Gaussian clustering algorithm are used. The incremental improvement due to either correlated-feature streams or the modified k -means Gaus-

sian clustering algorithm alone is similar in the case of CI models. In the case of CD models, most of the gain in accuracy comes from the modified k -means Gaussian clustering algorithm. Nonetheless, the improvements are observed with both CI and CD models at almost all levels of quantization — various numbers of subspace Gaussian prototypes. This shows that by bringing more knowledge into play — correlation in the correlated-feature stream definition and second-order statistics in the modified k -means Gaussian clustering algorithm, better subspace Gaussian tying is achieved.

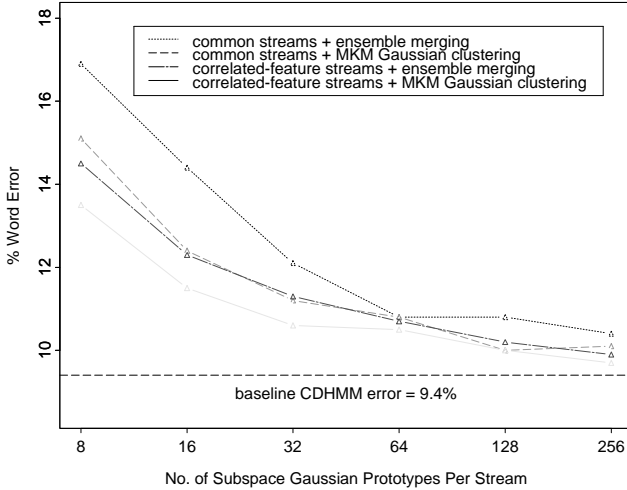
Henceforth, all experiments are run with SDCHMMs derived using the modified k -means Gaussian clustering algorithm with correlated-feature streams except for the 4-stream SDCHMMs which are derived with the common 4-stream definition.

C.2 II. Recognition Accuracy

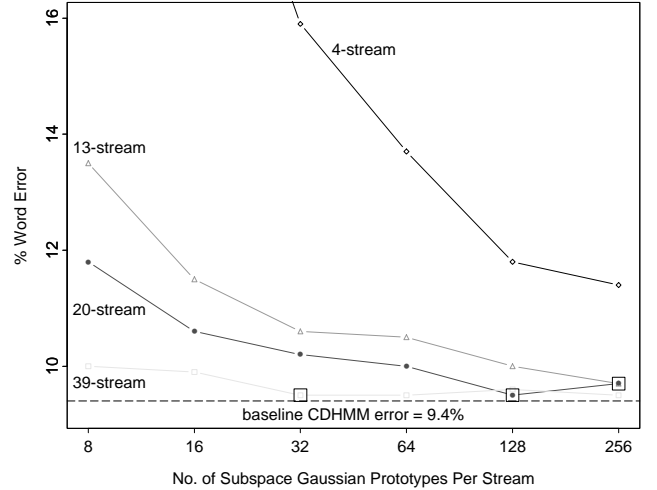
The baseline CI (CD) CDHMMs are converted to CI (CD) SDCHMMs with 8–256 (2–256) subspace Gaussian prototypes per stream. One, 4, 13, 20, and 39 streams are tried. Figure 4 shows their recognition accuracies in terms of word error rate (WER).

In general, WER decreases with more streams and more prototypes as expected, since more streams of smaller dimensions should result in smaller distortions when the subspace Gaussians are quantized, and more prototypes should give smaller quantization errors. For example, 39-stream CD SDCHMMs obtain the best WER of 5.0% with 16 subspace Gaussian prototypes, while 20-stream CD SDCHMMs require 64 prototypes, and 13-stream CD SDCHMMs reach their best WER of 5.2% with at least 128 prototypes. The best CI SDCHMMs (with 20 streams and 128 prototypes, or 39 streams and 32 prototypes) compare well with

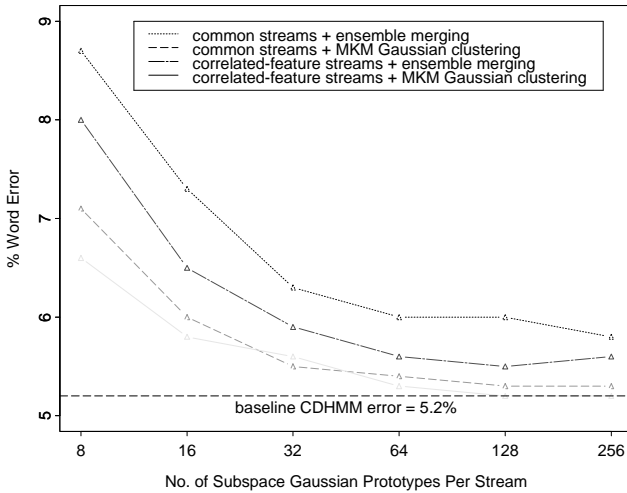
²We have also tried to compute the subspace Gaussian log-likelihoods on the fly during decoding, but unless when there are more than 512 prototypes per stream, pre-computation of the log-likelihoods always entails faster recognition.



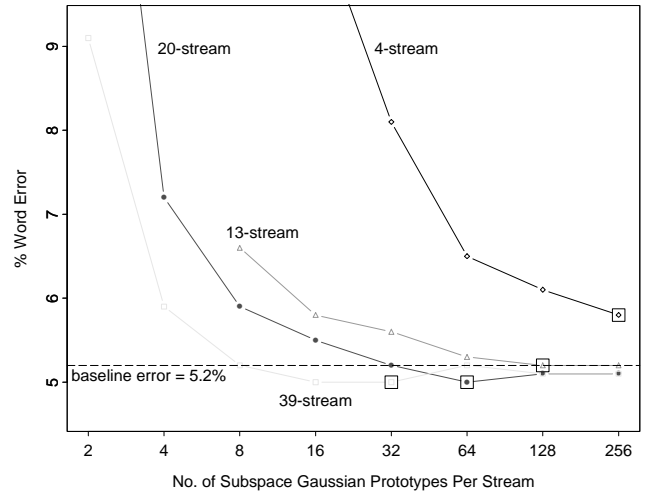
(a) Context-independent models



(a) Context-independent models



(b) Context-dependent models



(b) Context-dependent models

Fig. 3: ATIS: Recognition accuracy of 13-stream figure SDCHMMs with various stream definitions and clustering schemes

Fig. 4: ATIS: Effect of number of streams and subspace Gaussian prototypes on SDCHMM recognition accuracy (the best systems of Table III are marked with squares)

the baseline CI CDHMMs (9.5% vs. 9.4%), and the best CD SDCHMMs (with 20 streams and 64 prototypes, or 39 streams and 16 prototypes) actually outperform the baseline CD CDHMMs (5.0% vs. 5.2%). This suggests that some of the original CD CDHMMs may not be well trained, and subspace Gaussian tying may help improve these poor models by interpolating them with the better-trained models, or by pooling together more training data for them.

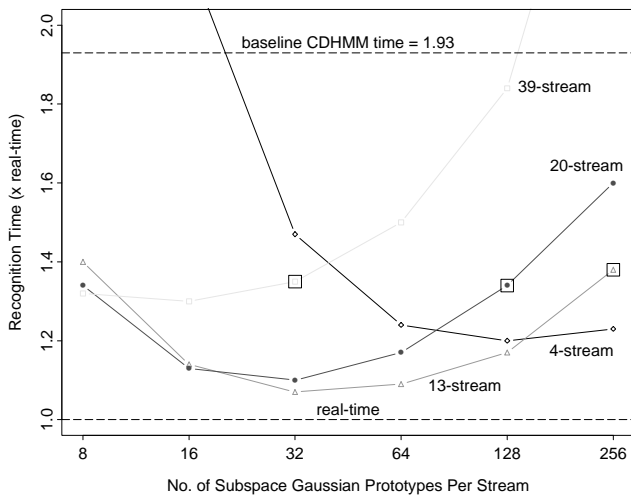
C.3 II. Recognition Speed

The corresponding total recognition times of the SDCHMM systems of Figure 4 are presented in Figure 5 relative to real-time performance. The relationships between

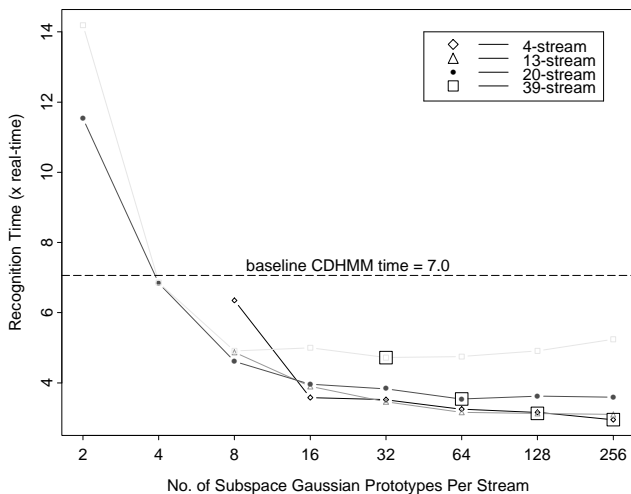
recognition speed and the number of prototypes are generally parabolas that curve upwards. The longer recognition time at the two ends of the parabolic curves are due to two very different effects:

- More prototypes simply require more computation for the subspace Gaussian log-likelihoods.
- Fewer prototypes lead to poorer SDCHMMs (due to larger quantization errors) with less discriminating power and more active states during a Viterbi search (using the same beam-width), and thus more computation.

The CD SDCHMM system is quite insensitive to the first effect when compared with the CI SDCHMM system. It is



(a) Context-independent models



(b) Context-dependent models

Fig. 5: ATIS: Effect of number of streams and sub-figure space Gaussian prototypes on SDCHMM recognition speed (the best systems of Table III are marked with squares)

because there are about 10 times more active states during decoding in the CD system. With the large number of active states in the CD system, the pre-computation of subspace Gaussian log-likelihoods represents a small proportion of the total computation time.

The impact of the number of streams on recognition speed is complicated by the above two effects, but in general, more streams means more additions in the computation of state log-likelihoods (Equation (5)) and more (software) function calls, hence longer recognition time.

C.4 Summary of Best Results

From the discussion above, there is a trade-off between recognition accuracy and recognition speed by adjusting the number of streams and the number of prototypes. By overlaying Figure 5 onto Figure 4, the best SDCHMM recognition systems with various numbers of streams are determined and summarized in Table III.

TABLE III: ATIS: Summary of the best results ($K = \#streams$, $n = \#subspace$ Gaussian prototypes per stream, CI = context independent, CD = context dependent, WER = word error rate (%), TIME is relative to that of the baseline system, PR = parameter reduction, and MS = memory savings. For PR, figures in parentheses take into account the mappings of subspace Gaussians to the full-space Gaussians. For MS, 1-byte mappings are assumed.)

CI/CD	K	n	WER	TIME	PR	MS
CI	1	2254	9.4	1.00	1	1
CI	13	256	9.7	0.72	8 (3.5)	6.1
CI	20	128	9.5	0.70	15 (3.1)	7.6
CI	39	32	9.5	0.70	38 (1.9)	6.7
CD	1	76154	5.2	1.00	1	1
CD	4	256	5.8	0.42	63 (15)	35
CD	13	128	5.2	0.44	70 (5.6)	18
CD	20	64	5.0	0.50	74 (3.8)	13
CD	39	32	5.0	0.67	77 (2.0)	7.3

The CD SDCHMMs perform better than the CI SDCHMMs when compared with their respective baseline systems. The CD SDCHMMs require fewer prototypes but give relatively better accuracies, higher computation efficiency, greater memory savings and larger reduction in model parameters. The most plausible explanation is that the CI models are less complex and robustly trained due to the large amount of available training data. Further tying of CI model parameters renders over-smoothing of the parameters. As a result, more prototypes are required to maintain acceptable quantization errors. On the contrary, the CD SDCHMMs are highly complex, and modeling the rare triphones has always been a problem. Obviously, results of Table III suggest that some triphones are still not well trained, and further tying at the smaller sub-phonetic unit of subspace Gaussians can effectively reduce the model parameter space to obtain more robust models. Nevertheless, it is still amazing to see that the 76,154 Gaussians of the baseline context-dependent CDHMMs can be represented by 32 — 128 subspace Gaussians per stream.

Thirteen, 20, or 39 streams all work well in both CD or CI systems, but their impacts on savings in computation, memory, model parameters and accuracy are quite different. For the CI systems, 13- to 39-stream SDCH-

MMs all give similar performance in terms of accuracy, speed and memory requirement. The only difference lies in their number of model parameters: 39-stream SDCHMMs (with 1-dimensional scalar streams) have the fewest model parameters if one does not count the subspace Gaussian encoding parameters, thanks to the efficiency of scalar quantization which requires fewer prototypes. However, once we include the encoding parameters, 39-stream SDCHMMs require more model parameters than SDCHMMs with fewer streams because they consume one encoding parameter per stream for each subspace Gaussian. On the other hand, since there are many more distributions and HMM state evaluations in CD systems than in CI systems, the greater sharing of Gaussian parameters in CD SDCHMMs entails greater savings in computation, memory, and model parameters.

Various statistical significance tests from NIST (National Institute of Standards and Technology) are run on the performance differences among the recognition systems of Table III. Most of the tests indicate no significant difference among the various CI(CD) systems. The only test that indicates a difference actually finds the SDCHMM systems more accurate.

C.5 Operating Curves

The foregoing discussion that is based on Viterbi decoding using one particular beam-width can be biased. Figure 6 studies the effect of beam-width on various SDCHMM systems of Table III with their operating curves.

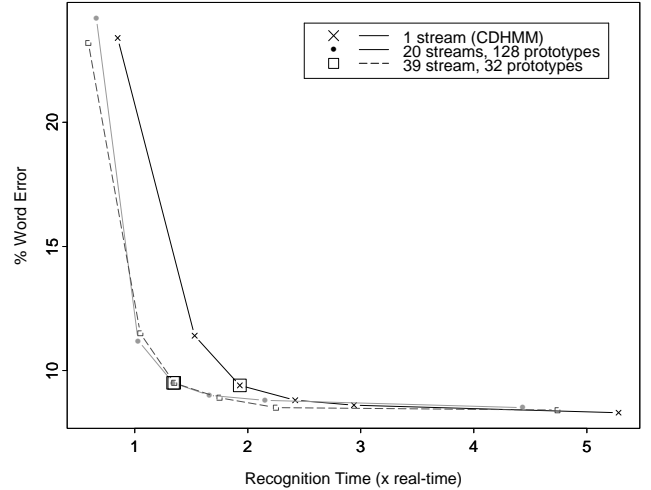
The asymptotic performances of CI SDCHMMs are basically the same as those of their parent CI CDHMMs, while CD SDCHMMs outperform CD CDHMMs asymptotically. In addition, the SDCHMM curves always lie to the left of the CDHMM curve on each graph; thus SDCHMM systems are always faster. Similarly, operating curves of SDCHMMs with fewer streams also lie to the left of SDCHMMs with more streams though they may saturate sooner with poorer accuracies (for example, compare the operating curves of 20-stream and 39-stream CI SDCHMMs, or those of 13-stream and 20-stream CD SDCHMMs). The best compromise seems to come from 20-stream SDCHMM systems.

V. COMPARISON WITH OTHER HMMs

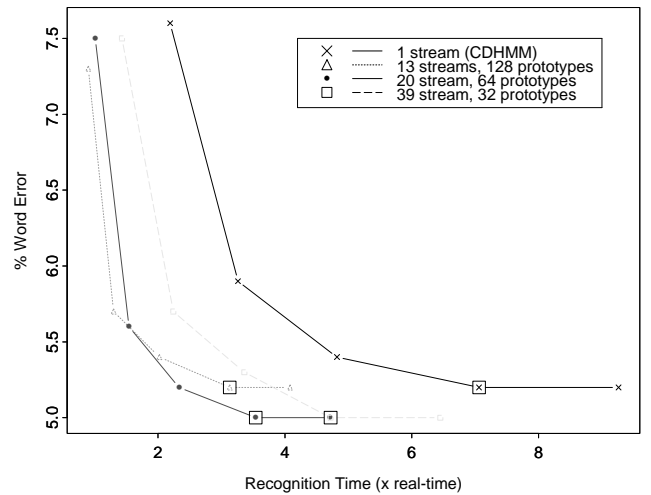
Our SDCHMM is very similar to two other hidden Markov modeling methodologies: semi-continuous HMM (SCHMM) [12], [13], [14] and feature-parameter-tying HMM (FPTHMM) [16], [15].

A. With Semi-Continuous HMM

At first glance, SDCHMM may appear similar to SCHMM: Both methods divide the feature space into streams, and tie subspace distributions across all states of all HMMs. However, close scrutiny shows that K -stream SCHMMs compute the state likelihood differently as



(a) Context-independent models



(b) Context-dependent models

Fig. 6: ATIS: Operating curves of SDCHMMs (the figure best systems of Table III are marked with squares)

$$P_{SCHMM}(\mathcal{O}) = \prod_{k=1}^K \left(\sum_{m=1}^M c_{mk} \mathcal{N}^{tied}(\mathcal{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2) \right) \quad (12)$$

where c_{mk} is the weight of the m -th mixture component in the k -th stream satisfying the stochastic constraint $\sum_{m=1}^M c_{mk} = 1$.

Comparing Equation (12) with Equation (5), one finds two differences:

- There is a switch between the product operator (\prod) and summation operator (\sum) in the two equations.
- In an SCHMM state, each of the K subspace Gaussians is associated with its own mixture weight c_{mk} , whereas

one mixture weight c_m is shared among all the K subspace Gaussians of a SDCHMM state.

Both differences arise from the fact that SCHMMs assume stream independence in the *global* feature space, whereas SDCHMMs assume stream independence in the *local* feature space — an assumption inherited from CDHMMs with mixture Gaussian densities and diagonal covariances. That is, for each state, SCHMMs estimate one mixture Gaussian density from each of the streams *independently*, and then combine the subspace Gaussian likelihoods by assuming again independent streams. However, the assumption of feature independence between the streams commonly used in speech recognition is hardly justified. SDCHMMs therefore start with CDHMMs using the full feature speech vectors without assuming any feature independence. The correlation between features at each state is well modeled by a mixture Gaussian density. An implication of the difference in the scope of the assumptions is the number of streams required: The SCHMM favors fewer streams of higher dimensions, so that correlation among more features can be modeled and there will be fewer mixture weights; on the contrary, SDCHMM favors more streams of lower dimensions so that quantization of the subspace Gaussians of CDHMMs will give smaller quantization errors and more accurate models.

Another difference between SDCHMM and SCHMM not readily observed from Equations (5) and (12) is that SCHMM requires each state to have the same number of mixture components equal to the number of distribution prototypes while SDCHMM does not. As a result, SDCHMM usually has many fewer mixture components per state, and thus has the following advantages over SCHMM:

- Fewer components mean fewer mixture weights which then take less memory space.
- Fewer components are involved in state likelihood computation which then takes less CPU time.

From the evaluation result in Section IV, the best SDCHMM systems have 13 or 20 streams with 32–128 subspace Gaussian prototypes. In contrast, from the literature, SCHMM systems usually require 64–512 prototypes using 3 or 4 streams. If these figures are representative, SDCHMM should require less model memory than SCHMM.

B. With Feature-Parameter-Tying HMM

The feature-parameter-tying HMM turns out to be a special case of our SDCHMM when the number of streams, K , is set to the size of the feature vector, D . In a sense, the FPTHMM is the scalar quantization (SQ) version of our SDCHMM. However, we note that,

1. the main storage cost of SDCHMMs is incurred by the subspace Gaussian encoding indices which grow in proportion with the number of streams; and,
 2. the computational cost of the state log-likelihood (Equation (5)) is directly proportional to the number of streams once all subspace Gaussian likelihoods are pre-computed.
- Thus, although SQ of the subspace Gaussians in FPTHMMs has the advantage of simplicity and generally gives the highest compression of subspace Gaussians, it needs

more storage space and more computation time than SDCHMMs with $K < D$. The difference is more conspicuous for large systems.

The evaluation results of Section IV, for example, Figure 6, have confirmed this.

VI. SUMMARY AND CONCLUSION

Continuous density hidden Markov modeling has been a milestone in the advancement of automatic speech recognition. However, its accuracy is achieved at the expense of high computational cost. In this paper, we show that subspace distribution clustering hidden Markov modeling can produce acoustic models that are as accurate as the CDHMMs, and yet they are much more compact. For example, on the ATIS task, compared with the baseline CDHMM system, the best context-dependent (context-independent) SDCHMM system saves the total computation time by 50% (30%) and obtains a 13-fold (8-fold) reduction in HMM memory with a relative 4% gain (1% drop) in accuracy.

SDCHMMs can be converted from a set of CDHMMs by properly projecting the mixture Gaussians of the CDHMMs onto subspaces, and carefully tying the ensuing subspace Gaussians. We propose to put the most correlated features into a stream. This correlated-feature stream definition, though not guaranteed optimal, is shown empirically giving good results. A modified k -means Gaussian clustering algorithm is also devised to tie the subspace Gaussians.

The CD SDCHMMs show greater relative improvements than the CI SDCHMMs probably due to the higher degree of redundancy and decreased robustness of the CD CDHMMs. One may thus postulate that SDCHMMs may be more effective with larger acoustic models.

The impact of the number of streams on accuracy, computation time, and memory size is complicated. All things considered, 13 and 20 streams seem to be better choices.

Re-training of the converted CI SDCHMMs has also been studied, and no significant improvement is observed. Since the converted CD SDCHMMs already surpass the baseline performance and training CD models requires a lot of fine-tuning, we thus did not attempt to re-train the CD SDCHMMs. Based on our experience with re-training the CI SDCHMMs, we are not surprised that re-training will not improve the CD SDCHMMs.

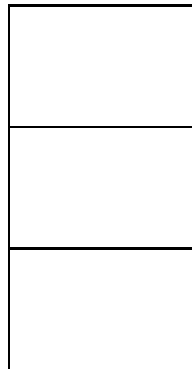
With the great reduction of Gaussian parameters (mixture weights, Gaussian means, and variances) by one to two orders of magnitude, one should expect SDCHMMs to be trained from scratch with much less training data than their parent CDHMMs. It should also be easier to adapt these fewer parameters for a new speaker or to another environment. These open exciting directions for our future work.

REFERENCES

- [1] M.K. Ravishankar, *Efficient Algorithms for Speech Recognition*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, 1996.

- [2] P. Beyerlein and M. Ullrich, "Hamming Distance Approximation for a Fast Log-Likelihood Computation for Mixture Densities," in *Proceedings of the European Conference on Speech Communication and Technology*, 1995, vol. 2, pp. 1083-1086.
- [3] E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993, vol. 2, pp. 692-695.
- [4] Y. Komori, M. Yamada, H. Yamamoto, and Y. Ohora, "An Efficient Output Probability Computation for Continuous HMM Using Rough and Detail Models," in *Proceedings of the European Conference on Speech Communication and Technology*, 1995, vol. 2, pp. 1087-1090.
- [5] M. Padmanabhan, D. Nahamoo L.R. Bahl, and P. de Souza, "Decision-Tree Based Quantization of the Feature Space of a Speech Recognizer," in *Proceedings of the European Conference on Speech Communication and Technology*, 1997, pp. 147-150.
- [6] F. Seide, "Fast Likelihood Computation for Continuous-Mixture Densities Using a Tree-Based Nearest Neighbor Search," in *Proceedings of the European Conference on Speech Communication and Technology*, 1995, vol. 2, pp. 1079-1082.
- [7] P.S. Gopalakrishnan and L.R. Bahl, "Fast Match Techniques," in *Automatic Speech and Speaker Recognition (Advanced Topics)*, C.H. Lee, F.K. Soong, and K.K. Paliwal, Eds., chapter 17, pp. 413-428. Kluwer Academic Publishers, 1996.
- [8] K.F. Lee and H.W. Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition Using HMM," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 123-126.
- [9] K.F. Lee, S. Hayamizu, H.W. Hon, C. Huang, J. Swartz, and R. Weide, "Allophone Clustering for Continuous Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, vol. 2, pp. 749-752.
- [10] M. Hwang, "Shared Distribution Hidden Markov Models for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 414-420, October 1993.
- [11] S.J. Young and P.C. Woodland, "The Use of State Tying in Continuous Speech Recognition," in *Proceedings of the European Conference on Speech Communication and Technology*, 1993, vol. 3, pp. 2203-2206.
- [12] 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, December 1990.
- [13] X. Huang and M.A. Jack, "Semi-continuous Hidden Markov Models for Speech Signals," *Journal of Computer Speech and Language*, vol. 3, no. 3, pp. 239-251, July 1989.
- [14] E. Singer and R.P. Lippmann, "A Speech Recognizer Using Radial Basis Function Neural Networks in an HMM Framework," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992, vol. 1, pp. 629-632.
- [15] S. Takahashi and S. Sagayama, "Four-Level Tied-Structure for Efficient Representation of Acoustic Modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, vol. 1, pp. 520-523.
- [16] S. Takahashi and S. Sagayama, "Effects of Variance Tying for Four-Level Tied Structure Phone Models," in *Proceedings of ASI Conference*, 1995, vol. 1-Q-23, pp. 141-142.
- [17] K.W. Law and C.F. Chan, "Split-Dimension Vector Quantization of Parcor Coefficients for Low Bit Rate Speech Coding," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 443-446, July 1994.
- [18] K.K. Paliwal and B.S. Atal, "Efficient Vector Quantization of LPC Parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3-14, January 1993.
- [19] B.H. Juang, D.Y. Gray, and A.H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 2, pp. 307-309, April 1982.
- [20] Yunxin Zhao, "A Speaker-Independent Continuous Speech Recognition System Using Continuous Mixture Gaussian Density HMM of Phoneme-Sized Units," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 3, pp. 345-361, July 1993.
- [21] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260-269, April 1967.
- [22] X.D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, 1990.
- [23] L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
- [24] S.K. Kachigan, *Multivariate Statistical Analysis (A Conceptual Introduction)*, Radius Press, 1991.
- [25] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithm*, Computer Science Press, 1978.
- [26] E. Bocchieri and G. Riccardi, "State Tying of Triphone HMM's for the 1994 AT&T ARPA ATIS Recognizer," in *Proceedings of the European Conference on Speech Communication and Technology*, 1995, vol. 2, pp. 1499-1502.
- [27] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc., 2nd edition, 1990.
- [28] T. Kosaka and S. Sagayama, "Tree-structured Speaker Clustering for Fast Speaker Adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994, vol. 1, pp. 245-248.
- [29] P.C. Loizou and A.S. Spanias, "High-Performance Alphabet Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 6, pp. 430-445, November 1996.
- [30] B. Mak and E. Barnard, "Phone Clustering Using the Bhattacharyya Distance," in *Proceedings of the International Conference on Spoken Language Processing*, 1996, vol. 4, pp. 2005-2008.
- [31] L. Hirschman et al., "Multi-Site Data Collection and Evaluation in Spoken Language Understanding," in *Proceedings of ARPA Human Language Technology Workshop*. 1993, Morgan Kaufmann Publishers.
- [32] D.S. Pallett et al., "1994 Benchmark Tests for the ARPA Spoken Language Program," in *Proceedings of ARPA Human Language Technology Workshop*. 1995, pp. 5-36, Morgan Kaufmann Publishers.

Enrico Bocchieri



Brian Kan-Wing Mak received the B.Sc. degree in Electrical Engineering from the University of Hong Kong in 1983, the M.S. degree in Computer Science from the University of California, Santa Barbara, USA in 1989, and the Ph.D. degree in Computer Science from Oregon Graduate Institute of Science and Technology, Portland, Oregon, USA in 1998. From 1990 until 1992, he was a research programmer at the Speech Technology Laboratory of Panasonic Technologies Inc. in Santa Barbara and worked on endpoint detection research in noisy environment. From 1997 until his Ph.D. graduation in 1998, he was also a research consultant at the AT&T Labs - Research, Florham Park, New Jersey, USA. Since April 1998, he has been an assistant professor at the Department of Computer Science, the Hong Kong University of Science and Technology. His interests include speech recognition, spoken language understanding, dialog modelling, and machine learning.