Subspace Distribution Clustering Hidden Markov Model

Enrico Bocchieri and Brian Kan-Wing Mak, *Associate 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 the parameters of a new unit, namely the subspace distribution, of continuous density hidden Markov models (CDHMMs). 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 Airline Travel Information System (ATIS) task shows that in comparison to its parent CDHMM system, a converted SDCHMM system achieves seven- to 18-fold reduction in memory requirement for acoustic models, and runs 30%–60% faster without any loss of recognition accuracy.**

***Index Terms*—Distribution clustering, hidden Markov modeling, subspace distribution.**

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 design these recognizers so that they may be run on more affordable machines of lower processing power and smaller memory size *without* losing accuracy. In the literature, there are 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 [1]–[8], or by fast-match techniques [9]. Another approach is to reduce the number of parameters in the acoustic models, to achieve the seemingly conflicting goals of

- high recognition accuracy;
- faster recognition;
- smaller memory requirement;
- requiring fewer training or adaptation data.

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 [10], generalized biphones/triphones [11]), states (tied-state HMM [12], [13]), observation distributions (tied-mixture/semicontinuous HMM [14]–[17]), and feature parameters [18] 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 unit—subspace (stream) distribution—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).

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 streams[1]), and similar stream distributions are then tied into a small number of distribution prototypes over *all* states and *all* acoustic models in each stream. In this study, clustering (of the CDHMM Gaussian projections) defines the tied subspace distributions. In [20] we would show that the parameters of these subspace distributions can be reestimated from speech data, according to maximum likelihood, using the expectation-maximization (EM) algorithm [21]. By 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.

[1]In 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.

Manuscript received June 29, 1998; revised April 27, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yunxin Zhao.

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

B. K.-W. Mak was with the Oregon Graduate Institute of Science and Technology, Portland, OR 97201 USA and also with AT&T Labs—Research, Florham Park, NJ 07932 USA. He is now with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: mak@cs.ust.hk).

Publisher Item Identifier S 1063-6676(01)01789-8.

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 [22], [18] 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" [23], [24] 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 [25]. 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. Subvector quantization for efficient speech recognition has recently been studied [26].

The above references and also this paper study tying of HMM parameters at different levels (i.e., tying of HMM states, Gaussians, etc.), however the actual number of model parameters is typically chosen by experiment or by other heuristics. Other recent studies have used model selection criteria from the statistics literature to determine the number of Gaussian components in acoustic models [27]–[29], for a given amount of training data.

The organization of this paper is as follows. In Section II, we present the concept of SDCHMM. Section III describes an implementation method in which SDCHMMs are converted from CDHMMs through Gaussian clustering algorithms. An algorithm for the definition of the streams based on feature correlation is also proposed. The SDCHMMs are 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):

$\boldsymbol{O}$	observation vector of dimension D;
$P(\boldsymbol{O})$	state output probability given $\boldsymbol{O}$;
c_{sm}	weight of the mth mixture component for the sth state;
$\boldsymbol{\mu}_{sm}$	mean vector of the mth mixture component for the sth state;
$\boldsymbol{\sigma}^2_{sm}$	variance vector of the mth component for the sth state;
$\mathcal{N}(\cdot)$	Gaussian pdf.

The observation probability density of state s is given by

$$P_s^{\mathrm{CDHMM}}(\boldsymbol{O}) = \sum_{m=1}^{M_s} c_{sm}\mathcal{N}(\boldsymbol{O};\, \boldsymbol{\mu}_{sm}, \boldsymbol{\sigma}^2_{sm}), \quad \sum_{m=1}^{M_s} c_{sm} = 1. \tag{1}$$

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, $1 \le k \le K$, with the following conditions.

Condition 1:

$$\sum_{k=1}^{K} d_k = D. \tag{2}$$

Condition 2:

$$\mathcal{R}^{d_i} \cap \mathcal{R}^{d_j} = \emptyset, \qquad 1 \le i \ne j \le K. \tag{3}$$

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 streams to obtain K subspace Gaussians of dimension d_k, $1 \le k \le K$, with diagonal covariances. That is, (1) can be rewritten as

$$P_s^{\mathrm{CDHMM}}(\boldsymbol{O}) = \sum_{m=1}^{M_s} c_{sm} \left(\prod_{k=1}^{K} \mathcal{N}(\boldsymbol{O}_k;\, \boldsymbol{\mu}_{smk}, \boldsymbol{\sigma}^2_{smk}) \right) \tag{4}$$

where $\boldsymbol{O}_k$, $\boldsymbol{\mu}_{smk}$, and $\boldsymbol{\sigma}^2_{smk}$ are the projection of the observation $\boldsymbol{O}$, and mean and variance vectors of the mth mixture component of the sth state onto the kth stream, respectively.

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

$$P_s^{\mathrm{SDCHMM}}(\boldsymbol{O}) = \sum_{m=1}^{M_s} c_{sm} \left(\prod_{k=1}^{K} \mathcal{N}^{tied}(\boldsymbol{O}_k;\, \boldsymbol{\mu}_{smk}, \boldsymbol{\sigma}^2_{smk}) \right). \tag{5}$$

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

The SDCHMM formulation can be generalized to any mixture density if the component pdf can be expressed as a product of subspace pdfs of the same functional form, provided that the three above conditions are satisfied. An obvious generalization is the mixture of Gaussians with block-diagonal covariances.

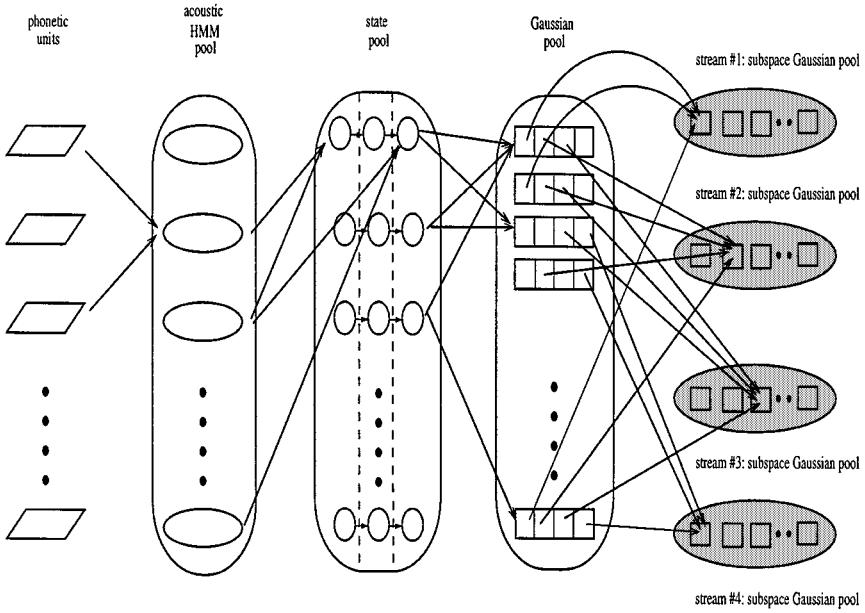


Fig. 1. Subspace distribution clustering hidden Markov models with four streams.

However we investigate only SDCHMMs based on CDHMMs with mixture Gaussian densities and diagonal covariances.

B. Distribution Clustering

In practice, the proposed SDCHMM as in (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 prototypes

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

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

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

with l being given by

$$l = \arg \min_{1 \le q \le L} dist\left(\mathcal{N}(\boldsymbol{O}_k; \boldsymbol{\mu}_{smk}, \boldsymbol{\sigma}^2_{smk},) \cdot \mathcal{N}^{quantized}(\boldsymbol{O}_k; \boldsymbol{\mu}_{qk}, \boldsymbol{\sigma}^2_{qk})\right) \quad (6)$$

where $dist(\cdot)$ 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 two 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 precomputed once and only once at the beginning of every frame, and their values are stored in lookup tables. During Viterbi decoding [31] of a K-stream SDCHMM system, the log likelihood of a Gaussian component of a state can be computed as the summation of K precomputed 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 (5) of Section II suggests that SDCHMMs may be implemented in the following two steps as shown in Fig. 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 (or stream) Gaussians in each stream.

Since the training of CDHMMs is well covered in the literature [32], [33], 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 [34], [35].

Tying of subspace Gaussians consists of splitting the full speech feature vector space into disjoint subspaces (or streams), projecting mixture Gaussians of CDHMMs onto these subspaces, and then clustering the subspace Gaussians into a small number of Gaussian prototypes in each subspace.

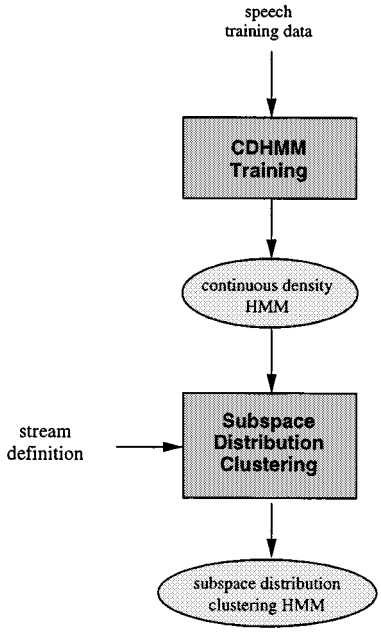


Fig. 2. Conversion of CDHMMs to SDCHMMs.

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, $1 \le k \le K$. Formally, let $\mathcal{P}_K^D$ be such a partition, then

$$\mathcal{P}_K^D = \left\{ \Omega^{d_k} \colon \sum_{k=1}^{K} d_k = D \quad \text{and} \quad \Omega^{d_k} \cap \Omega^{d_j} = \emptyset \right\} \tag{7}$$

where $1 \le k \ne j \le K$.

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 predetermined 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.

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 semicontinuous HMM, the following "common" definitions of streams are explored.

1-Stream Definition:

$$\boxed{12\text{MFCC} + 12\Delta\text{MFCC} + 12\Delta^2\text{MFCC} + e + \Delta e + \Delta^2 e}$$

4-Stream Definition:

$$\boxed{12\text{MFCC}}$$
$$\boxed{12\Delta\text{MFCC}}$$
$$\boxed{12\Delta^2\text{MFCC}}$$
$$\boxed{e + \Delta e + \Delta^2 e}$$

13-Stream Definition:

$$12 * \boxed{\text{MFCC} + \Delta\text{MFCC} + \Delta^2\text{MFCC}}$$
$$\boxed{e + \Delta e + \Delta^2 e}$$

39-Stream Definition: each one-dimensional (1-D) 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.

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.

a) Multiple correlation measure: The correlation ρ_{ij} between two variables is commonly measured by Pearson's moment product correlation coefficient

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{8}$$

where σ_i and σ_j are the standard deviations of the ith and jth 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 [36]. 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

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

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

$$R = 1 - \begin{vmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1k} \\ \rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2k} \\ \rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k1} & \rho_{k2} & \rho_{k3} & \cdots & 1 \end{vmatrix}. \tag{9}$$

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., $\forall i, j, \rho_{ij} = 0$, $R = 0$.

b) Derivation of streams: Practically, we apply a greedy algorithm [37] 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 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.

Table I shows the definition of 20 correlated-feature streams generated by Algorithm 1 using 1000 utterances from the ATIS training corpus. From the definition, MFCC and Δ^2MFCC are found mostly correlated.

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 (9). [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.

B. Issue II—Subspace Gaussian Clustering

Two very different clustering schemes are investigated: A bottom-up agglomerative clustering algorithm [19] and a top-down modified k-means (MKM) clustering algorithm.

1) Agglomerative Gaussian Clustering Algorithm: The *ensemble merging algorithm* for state tying described in [38] 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 $O(n^2)$.

TABLE I
ARTIS: 20 CORRELATED-FEATURE STREAMS

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	Δc_1, Δc_7
14	Δc_2, Δc_6
15	Δc_3, Δc_5
16	Δc_4, e
17	Δc_8, Δc_9
18	Δc_{10}, Δc_{11}
19	Δe, $\Delta\Delta e$
20	Δc_{12}

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).

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

$$D_{bhat} = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \left[\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right]^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2} \ln \frac{\left|\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_1|\,|\boldsymbol{\Sigma}_2|}}. \quad (10)$$

Algorithm 2: Modified k-Means Gaussians Clustering Algorithm

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

TABLE II
ATIS: 20 TESTING CONDITIONS AND PERFORMANCE OF THE BASELINE CI/CD SYSTEMS

CONDITION/PERFORMANCE	CI SYSTEM	CD SYSTEM
#Test Sentences	981 (1994 ARPA-ATIS evaluation set)	
Vocabulary	1,536 words	
Language Model	word-sequence bigram (perplexity $\approx$ 20)	
#Training Utterances	~12,000 ATIS	~20,000 ATIS
#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

Step 1) Initialization: First train a one-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 Steps 4 and 5 until some convergence criterion is met;

Step 4) Membership: Associate each subspace Gaussian of CDHMMs with it 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.

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, $i = 1, 2$, are the means and covariances of the two Gaussians [39]. The Bhattacharyya distance has been used in several speech-related tasks [40]–[42], 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 1000 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 are available, one may, for example, randomly pick L subspace Gaussians from the CDHMMs to start the clustering procedure.

IV. EVALUATION OF SDCHMM

A. ATIS Task

The Air Travel Information System (ATIS) [43] 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 1536 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. Baseline CDHMM Recognizer

Our baseline system consists of AT&T's ATIS recognizer used in the 1994 ARPA–ATIS evaluation [44], [45]. 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 front-end is based on mel-frequency cepstral analysis of input speech sampled at 16 kHz. At every 10 ms, 31

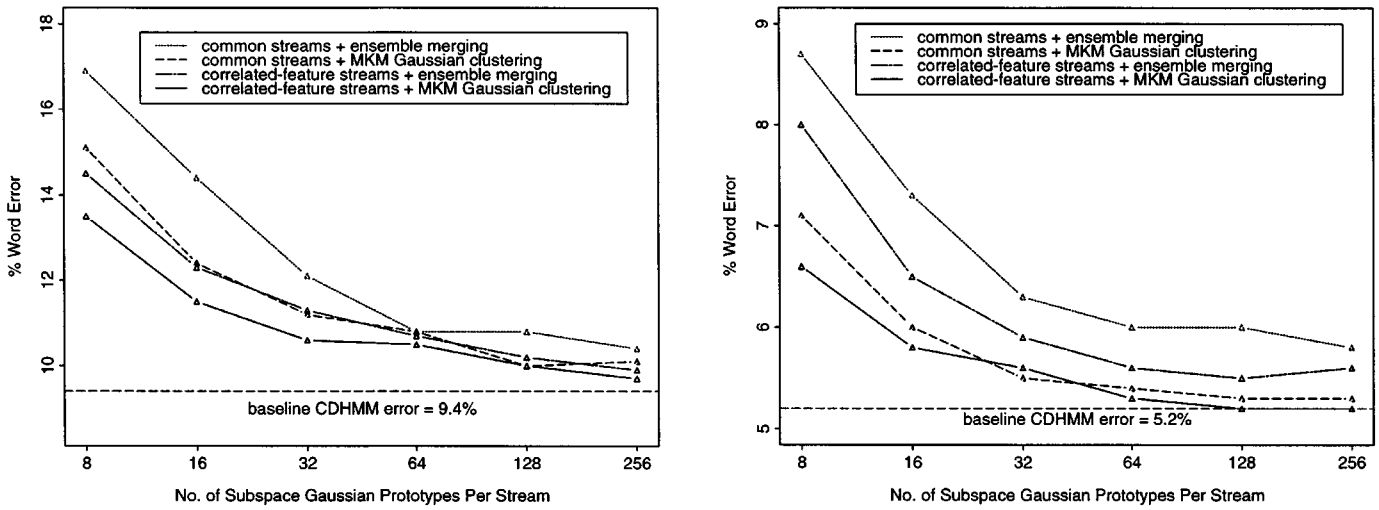


Fig. 3. ATIS: Recognition accuracy of 13-stream SDCHMMs with various stream definitions and clustering schemes. (a) Context-independent models and (b) context-dependent models.

mel-frequency energy components are computed from a filter bank by performing an FFT on a *frame* of 20 ms 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 CDHMMs to SDCHMMs. The testing conditions are exactly the same as those described in Table II. All subspace (stream) Gaussian log-likelihoods are precomputed at the beginning of each frame, and their values are stored in tables in contiguous memory.[2] In addition, for implementation and system simplicity, all streams are tied to the same number of subspace Gaussian prototypes in all our SDCHMMs.

1) 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 an identical number of subspace Gaussian prototypes, ranging from eight to 256 in different experiments. Each of the ensuing 13-stream SDCHMM systems is then tested on the 1994 ATIS evaluation dataset.

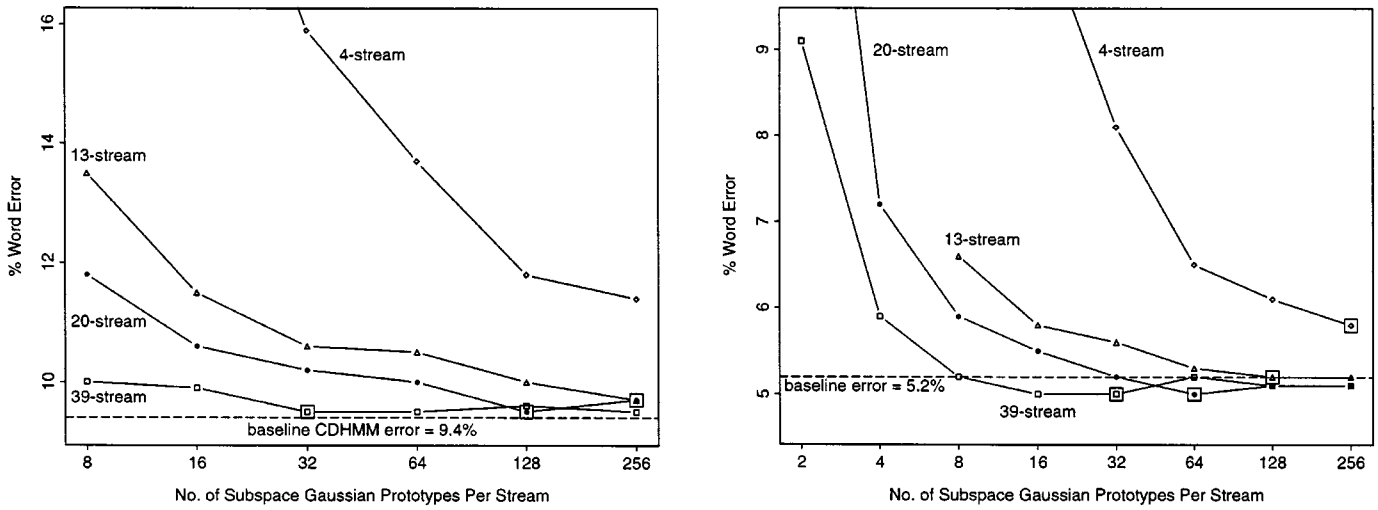
Fig. 3(a) and (b) show incremental improvements in recognition performance when correlated-feature streams and/or the modified k-means Gaussian clustering algorithm are used. The incremental improvement due to either correlated-feature streams or the modified k-means Gaussian clustering algorithm alone 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 four-stream SDCHMMs which are derived with the common four-stream definition.

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

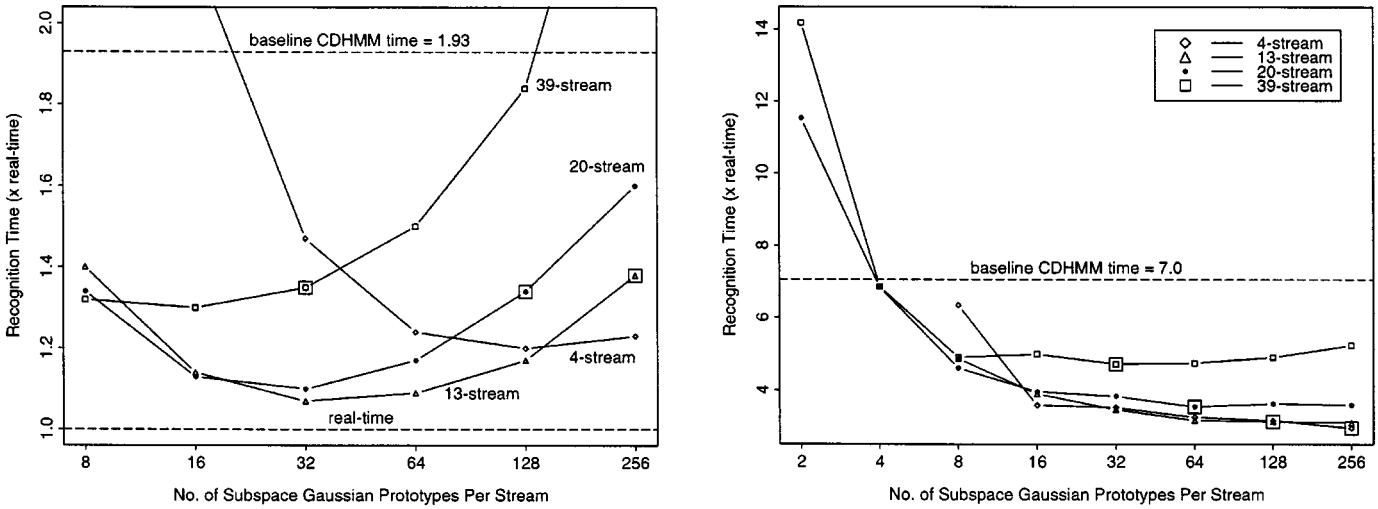
In general, WER decreases with more streams and more prototypes as expected, since more streams of smaller dimensions should result in smaller distortions when the subspace Gaussians are quantized, and more prototypes should give smaller

[2]We 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, precomputation of the log-likelihoods always entails faster recognition.



(a) (b)

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). (a) Context-independent models and (b) context-dependent models.



(a) (b)

Fig. 5. ATIS: Effect of number of streams and subspace Gaussian prototypes on SDCHMM recognition speed (the best systems of Table III are marked with squares). (a) Context-independent models and (b) context-dependent models.

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 the baseline CI CDHMMs (9.5% versus 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% versus 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.

3) Recognition Speed: The corresponding total recognition times of the SDCHMM systems of Fig. 4 are presented in Fig. 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 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

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

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

streams means more additions in the computation of state log-likelihoods (5) and more (software) function calls, hence longer recognition time.

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 Fig. 5 onto Fig. 4, the best SDCHMM recognition systems with various numbers of streams are determined and summarized in Table III.

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 (finer than state) 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 SDCHMMs 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-D 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 National Institute of Standards and Technology (NIST) are run on the performance

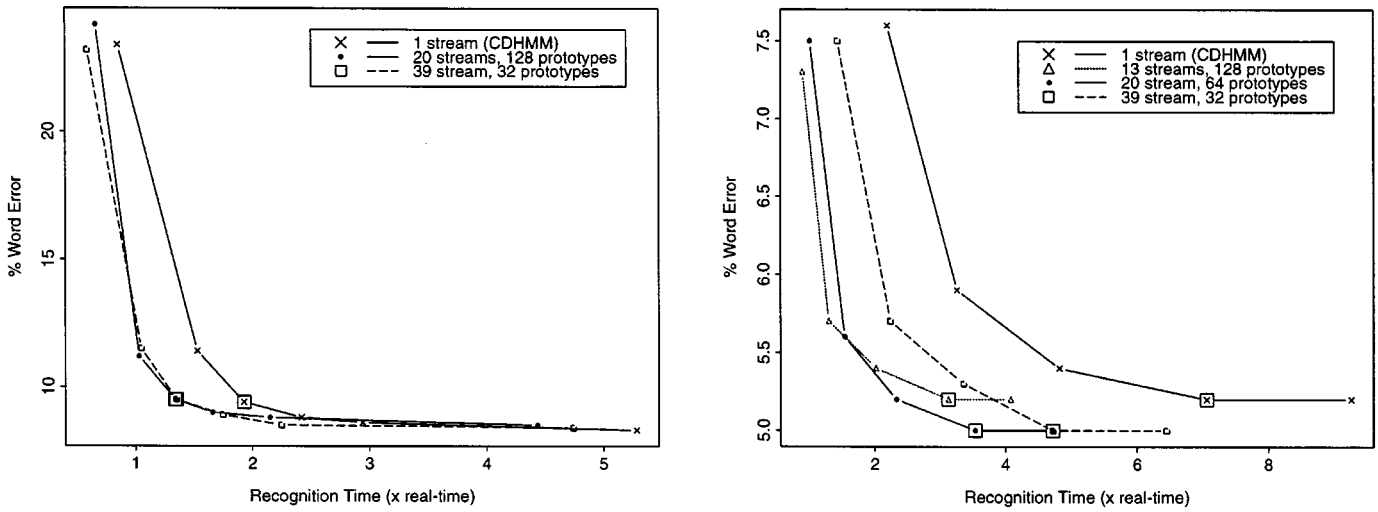


Fig. 6. ATIS: Operating curves of SDCHMMs (the best systems of Table III are marked with squares).

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.

5) Operating Curves: The previous discussion that is based on Viterbi decoding using one particular beam-width can be biased. Fig. 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

We compare our SDCHMM to two other hidden Markov modeling methodologies: semi-continuous HMM (SCHMM) [14], [15], and feature-parameter-tying HMM (FPTHMM) [18], [22].

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 (or stream) distributions across all states of all HMMs. However, close scrutiny shows that K-stream SCHMMs compute the state likelihood differently as

$$P_s^{\text{SCHMM}}(\boldsymbol{O}) = \prod_{k=1}^{K} \left(\sum_{m=1}^{M} c_{smk} \mathcal{N}^{tied}(\boldsymbol{O}_k; \boldsymbol{\mu}_{mk}, \boldsymbol{\sigma}_{mk}^2 \right) \tag{11}$$

where c_{smk} is the weight in the sth state of the mth mixture component in the kth stream satisfying the stochastic constraint $\sum_{m=1}^{M} c_{smk} = 1$.

Comparing (11) with (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_{smk}, whereas one mixture weight c_{sm} 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 state probability density function definition, while SDCHMMs do not. 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. Conversly, 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.

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) main storage cost of SDCHMMs is incurred by the subspace Gaussian encoding indices which grow in proportion with the number of streams;

2) computational cost of the state log-likelihood (5) is directly proportional to the number of streams once all subspace Gaussian likelihoods are precomputed.

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, Fig. 6, have confirmed this.

C. With Gaussian Selection

SDCHMM and Gaussian selection [1]–[7] achieve computation savings through two different principles. SDCHMM can be thought as an approximation of the CDHMM Gaussians, which achieves likelihood computation by tying parameters across subspace Gaussians. Gaussian selection is a pruning scheme that limits the likelihood computation to the most relevant Gaussians of the CDHMM. By applying Gaussian selection to the SDCHMM *approximations* of the full-space Gaussians, we have found that the computation savings of the two techniques are to some extent cumulative. The recognition times in Fig. 5 are with SDCHMM only. An additional 10% to 15% total computation time reduction was obtained together with Gaussian selection [2], and further savings should be obtained with more recent development of the Gaussian selection technique.

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 (or stream) 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 projecting the mixture Gaussians of the CDHMMs onto subspaces, and 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.

A direction for future study is whether the tying structure of the subspace Gaussians is reasonably consistent and portable across different applications and acoustic environments. In this case, 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.

References

[1] P. Beyerlein and M. Ullrich, "Hamming distance approximation for a fast log-likelihood computation for mixture densities," in *Proc. Eur. Conf. Speech Communication Technology*, vol. 2, 1995, pp. 1083–1086.

[2] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, 1993, pp. 692–695.

[3] Y. Komori, M. Yamada, H. Yamamoto, and Y. Ohora, "An efficient output probability computation for continuous HMM using rough and detail models," in *Proc. Eur. Conf. Speech Communication Technology*, vol. 2, 1995, pp. 1087–1090.

[4] M. Padmanabhan, D. Nahamoo, L. R. Bahl, and P. de Souza, "Decision-tree based quantization of the feature space of a speech recognizer," in *Proc. Eur. Conf. Speech Communication Technology*, 1997, pp. 147–150.

[5] F. Seide, "Fast likelihood computation for continuous-mixture densities using a tree-based nearest neighbor search," in *Proc. Eur. Conf. Speech Communication Technology*, vol. 2, 1995, pp. 1079–1082.

[6] S. H. Herman and R. A. Sukkar, "Joint MCE estimation of VQ and HMM parameters for Gaussian mixture selection," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1998, pp. 485–488.

[7] M. J. F. Gales, K. M. Knill, and S. J. Young, "State based Gaussian selection In large vocabulary continuous speech recognition using HMMs," *IEEE Trans. Speech Audio Processing*, vol. 7, 1999.

[8] D. B. Paul, "An investigation of Gaussian shortlists," in *Proc. 1999 IEEE Speech Recognition Understanding Workshop*, Keystone, CO, Dec. 1999.

[9] 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. New York: Kluwer Academic, 1996, ch. 17, pp. 413–428.

[10] K. F. Lee and H. W. Hon, "Large-vocabulary speaker-independent continuous speech recognition using HMM," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1988, pp. 123–126.

[11] K. F. Lee, S. Hayamizu, H. W. Hon, C. Huang, J. Swartz, and R. Weide, "Allophone clustering for continuous speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, 1990, pp. 749–752.

[12] M. Hwang, "Shared distribution hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 414–420, Oct. 1993.

[13] S. J. Young and P. C. Woodland, "The use of state tying in continuous speech recognition," in *Proc. Eur. Conf. Speech Communication Technology*, vol. 3, 1993, pp. 2203–2206.

[14] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 2033–2045, Dec. 1990.

[15] X. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *J. Comput. Speech Lang.*, vol. 3, pp. 239–251, July 1989.

[16] E. Singer and R. P. Lippmann, "A speech recognizer using radial basis function neural networks in an HMM framework," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, 1992, pp. 629–632.

[17] V. Digalakis and H. Murveit, "Genones: Optimizing the degree of mixture tying in a large vocabulary hidden Markov model based speech recognizer," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, 1994, pp. 537–540.

[18] S. Takahashi and S. Sagayama, "Four-level tied-structure for efficient representation of acoustic modeling," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, 1995, pp. 520–523.

[19] R. O. Duda and P. E. Hart, *Pattern Classification And Scene Analysis*. New York: Wiley, 1973.

[20] B. Mak and E. Bocchieri, "Direct training of the context-independent subspace distribution clustering hidden Markov model," *IEEE Trans. Speech Audio Processing*, to be published.
[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.
[22] S. Takahashi and S. Sagayama, "Effects of variance tying for four-level tied structure phone models," in *Proc. ASI Conf.*, vol. 1-Q-23, 1995, pp. 141–142.
[23] K. W. Law and C. F. Chan, "Split-dimension vector quantization of Parcor coefficients for low bit rate speech coding," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 443–446, July 1994.
[24] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 3–14, Jan. 1993.
[25] B. H. Juang, D. Y. Gray, and A. H. Gray, Jr. , "Distortion performance of vector quantization for LPC voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 307–309, Apr. 1982.
[26] S. Tsakalidis, V. Digalakis, and L. Neumeyer, "Efficient speech recognition using subvector quantization and discrete-mixture HMMs," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Mar. 1999, pp. 569–572.
[27] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 1998, pp. 645–648.
[28] L. R. Bahl and M. Padmanabhan, "A discriminant measure for model complexity adaptation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 1998, pp. 453–456.
[29] S. S. Chen, "Recent improvements to IBM's speech recognition system for automatic speech recognition," in *Proc. DARPA Speech Recognition Workshop*, 1999.
[30] Y. Zhao, "A speaker-independent continuous speech recognition system using continuous mixture Gaussian density HMM of phoneme-sized units," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 345–361, July 1993.
[31] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.
[32] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh, U.K.: Edinburgh Univ. Press, 1990.
[33] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[34] E. Bocchieri and B. Mak, "Subspace distribution clustering for continuous observation density hidden Markov models," in *Proc. 5th Eur. Conf. Speech Communication Technology*, vol. 1, Rhodes, Greece, Sept. 1997, pp. 107–110.
[35] B. Mak, E. Bocchieri, and E. Barnard, "Scheme derivation and clustering scheme for subspace distribution clustering hidden Markov model," in *Proc. IEEE ASRO Workshop*, Santa Barbara, CA, Dec. 1997, pp. 339–346.
[36] S. K. Kachigan, *Multivariate Statistical Analysis (A Conceptual Introduction)*. New York: Radius, 1991.
[37] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithm*. Rockville, MD: Computer Science, 1978.
[38] E. Bocchieri and G. Riccardi, "State tying of triphone HMM's for the 1994 AT&T ARPA ATIS recognizer," in *Proc. Eur. Conf. Speech Communication Technology*, vol. 2, 1995, pp. 1499–1502.
[39] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
[40] T. Kosaka and S. Sagayama, "Tree-structured speaker clustering for fast speaker adaptation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, 1994, pp. 245–248.
[41] P. C. Loizou and A. S. Spanias, "High-performance alphabet recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 430–445, Nov. 1996.
[42] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," in *Proc. Int. Conf. Spoken Language Processing*, vol. 4, 1996, pp. 2005–2008.
[43] L. Hirschman, "Multi-site data collection and evaluation in spoken language understanding," in *Proceedings of ARPA Human Language Technology Workshop*. San Mateo, CA, 1993.
[44] D. S. Pallett, "1994 Benchmark Tests for the ARPA Spoken Language Program," in *Proceedings of ARPA Human Language Technology Workshop*. San Mateo, CA, 1995, pp. 5–36.
[45] E. Bocchieri, G. Riccardi, and J. Anantharaman, "The 1994 AT&T ATIS CHRONUS recognizer," in *Proceedings of ARPA Human Language Technology Workshop*. San Mateo, CA: Morgan Kaufmann, 1995, pp. 265–268.

Enrico Bocchieri received the Laurea degree (with honors) in electrical engineering from the University of Pavia, Pavia, Italy, in 1979, and the M.S. and Ph.D. degrees in electrical enginnering from the University of Florida, Gainesville, in 1981 and 1983, respectively.

He was a Member of Technical Staff with the Central Research Laboratories, Texas Instruments, from 1984 to 1987, and with Bell Communication Research from 1987 to 1990. He then joined the Information Principle Research Laboratories, AT&T Bell Laboratories. He is now a Principal Technical Staff Member with the Speech and Image Processing Services Research Laboratory, AT&T Labs—Research, Florham Park, NJ. His research interests include speech recognition and understanding, signal processing, and software engineering.

Brian Kan-Wing Mak (S'96–A'98) 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, in 1989, and the Ph.D. degree in computer science from Oregon Graduate Institute of Science and Technology, Portland, in 1998.

From 1990 to 1992, he was a Research Programmer with the Speech Technology Laboratory, Panasonic Technologies, Inc., Santa Barbara, and worked on endpoint detection research in noisy environments. From 1997 to 1998, he was also a Research Consultant with AT&T Labs—Research, Florham Park, NJ. Since April 1998, he has been an Assistant Professor with the Department of Computer Science, Hong Kong University of Science and Technology. His interests include speech recognition, spoken language understanding, dialogue modeling, and machine learning.