Appears in ***Speech Communication*** **40 (2003) 145-149.**

Modeling Drivers' Speech Under Stress

Raul Fernandez, Rosalind W. Picard
{galt,picard}media.mit.edu
MIT Media Laboratory
20 Ames. St., Cambridge, MA 02139 – US

Abstract

We explore the use of features derived from multiresolution analysis of speech and the Teager Energy Operator for classification of drivers' speech under stressed conditions. We apply this set of features to a database of short speech utterances to create user-dependent discriminants of four stress categories. In addition we address the problem of choosing a suitable temporal scale for representing categorical differences in the data. This leads to two modeling approaches. In the first approach, the dynamics of the feature set *within* the utterance are assumed to be important for the classification task. These features are then classified using dynamic Bayesian network (DBN) models as well as a model consisting of a mixture of hidden Markov models (M-HMM). In the second approach, we define an utterance-level feature set by taking the mean value of the features *across* the utterance. This feature set is then modeled with a support vector machine and a multilayer perceptron classifier. We compare the performance on the sparser and full dynamic representations against a chance-level performance of 25% and obtain the best performance with the speaker-dependent mixture model (96.4% on the training set, and 61.2% on a separate testing set). We also investigate how these models perform on the speaker-independent task. Although the performance of the speaker-independent models degrades with respect to the models trained on individual speakers, the mixture model still outperforms the competing models and achieves significantly better than random recognition (80.4% on the training set, and 51.2% on a separate testing set).

Zusamenfassung

In diesem Bericht untersuchen wir die Verwendung von Merkmalen der Sprachanalyse aufgrund mehrerer Zeitskalen zur Klassifikation der Sprache eines Fahrers unter Stress. Diese Merkmale wenden wir auf eine Datenbank kurzer Sprachsequenzen an, um vier sprecherabhängige Stresskategorien zu erstellen. Zusätzlich beschäftigen wir uns mit der Auswahl der passenden Zeitskala für die Repräsentation klassenspezifischer Unterschiede in der Datenmenge. Dies führt zu zwei unterschiedlichen Modellierungsansätzen. Im ersten Ansatz wird vorausgesetzt, dass die dynamische Entwicklung der Merkmale, die innerhalb der Sprachsequenzen vorhanden ist, wichtig für die

Klassifizierung ist. Diese Merkmale werden klassifiziert mit Hilfe von dynamischen Bayes Netzen (DBN) bzw. mit einer Mischung von Hidden Markov Modellen. Im zweiten Ansatz definieren wir Merkmale auf Artikulationsebene, indem wir die Mittelwerte der Merkmale über die Sprachsequenzen berechnen. Diese Merkmalsmenge wird dann mit einer Support Vector Maschine und einem Multilayer-Perzeptron modelliert. Die Performanz der spärlichen und der voll dynamischen Darstellung wird verglichen mit dem zufälligen Klassifikationsniveau von 25%. Das beste Ergebnis erhalten wir hierbei mit einem sprecherabhängigen Mischmodell (96.4% auf den Trainingsdaten und 61.2% auf unabhängigen Testdaten). Weiterhin untersuchen wir, wie diese Modelle bei sprecherunabhängigen Aufgaben abschneiden. Obwohl die sprecherunabhängigen Modelle schlechter abschneiden als die auf einzelne Sprecher justierten Modellen, übertrifft das Mischmodell die konkurrierenden Modelle immer noch und erzielt signifikant bessere Sprechererkennung als Zufallserkennung (80.4% auf den Trainingsdaten und 51.2% auf unabhängigen Testdaten).

Résumé

Nous explorons l'utilisation de représentations dérivées de l'analyse multirésolution de la parole et de l'opérateur d'énergie de Teager pour la classification de la parole de conducteurs en condition de stress. Nous appliquons cette analyse à corpus d'énoncés courts pour créer des fonctions discriminantes dépendantes du locuteur pour quatre catégories de stress. En outre nous adressons le problème du choix d'une échelle temporelle appropriée pour catégoriser les données. Ceci mène à deux approches pour la modélisation. Dans la première approche, la dynamique des variables issues de l'analyse d'un énoncé donné est supposée pertinente pour la classification. Ces variables sont alors modélisées au moyen de réseaux bayésiens dynamiques (DBN) ou par un mélange des modèles de Markov cachés (M-HMM). Pour la seconde approche, nous ne gardons que les valeurs moyennes de ces variables pour chaque énoncé. Le vecteur résultant est alors modélisé au moyen d'une machine à support de vecteur et d'un perceptron multicouches. Nous comparons les performances de ces deux approches à un tirage aléatoire (25%), les meilleurs résultats étant obtenus avec le mélange de modèles dépendant du locuteur (96,4% sur les données d'apprentissage, et 61,2% sur un jeu de test distinct). Nous étudions également les performances de modèles indépendants du locuteur. Bien que les performances se dégradent par rapport des modèles spécifiques aux locuteurs, le mélange de modèles surpasse encore les autres modèles et obtient un taux de reconnaissance sensiblement meilleur qu'un tirage aléatoire (80,4% sur les données d'apprentissage, et 51,2% sur le jeu de test).

Keywords: Stress in speech; Automatic recognition.

1 Introduction

Much of the current effort on studying speech under stress has been aimed at detecting stress conditions for improving the robustness of speech recognizers; typical research of speech under stress has targeted perceptual (e.g. Lombard effect), psychological (e.g. timed tasks), as well as physical stressors (e.g. roller-coaster rides, high G forces) (Steeneken and Hansen, 1999). In this work we are interested in modeling speech in the context of driving under varying conditions of cognitive load hypothesized to induce a level of stress on the driver. The results of this research may be relevant not only to building recognition systems that are more robust in the context described, but also to applications that attempt to infer the underlying affective state of an utterance. We have chosen the scenario of driving while talking on the phone as an application in which knowledge of the driver's state may provide benefits ranging from a more fluid interaction with a speech interface to improvement of safety in the response of the vehicle.

The recent literature discussing the effects of stress on speech applies the label of *stress* to different phenomena. Some work views stress as any broad deviations in the production of speech from its normal production (Hansen and Womack, 1996; Sarikaya and Gowdy, 1998). In discussing the SUSAS database for the study of speech under stress, Hansen et al. (1998) go on to describe various types of stress in speech. These include the effects that speaking styles, noise, and G-forces have on the speaker's output, as well as the effect of states that are often described under the label of *emotions* elsewhere in the literature (e.g., anxiety, fear, or anger).

In an attempt to unify the often diverging views of stress that are being invoked by the research in this field, Murray et al. (1996) have reviewed various definitions of stress, and proposed a description of this phenomenon based on the character of the *stressors*. They have hypothesized four levels of stressors that can affect the speech production process. At the lowest level, these include direct changes on the vocal apparatus (zero-order stressors), unconscious physiological changes (first-order stressors), and conscious physiological changes (second-order stressors), At the highest level, changes can also be brought about by stimuli that are external to the speech production process, by the speaker's cognitive reinterpretation of the context in which the speech is being produced, as well as by the speaker's underlying affective conditions (third-order stressors). In this paper, we follow this taxonomy and investigate whether it is possible to discriminate between spoken utterances that have been produced under the influence of third-order stressors with a varying degree of stress.

2 Speech Corpus and Data Annotation

The speech data used in the research presented in this paper was collected from subjects driving in a simulator at the Nissan's Cambridge Research Lab. Subjects were asked to drive through a course while engaged in a simulated phone task: while the subject drove, a speech synthesizer prompted the driver with a math question consisting of adding up two numbers whose sum was less than 100. We controlled for the number of additions with and without carry-ons in order to maintain an approximately constant level of difficulty across trials.

The two independent variables in this experiment were the driving speed and the frequency with which the driver was given math questions. Each variable had two conditions – slow and fast – resulting in four different combinations. Subjects drove at 60 m.p.h. in the slow speed condition and at 120 m.p.h. in the fast speed condition (the perceptual speed in the simulator is approximately half). One subject complained of motion sickness in the fast speed condition, so in that case the speed was reduced to 100 m.p.h. The frequency at which drivers were prompted with a math question was once every 9 seconds (slow condition) or once every 4 seconds (fast condition). The driver's answers were captured by a head-mounted microphone and recorded in VHS format. The corpus analyzed here consists of 598 utterances (154, 156, 137 and 151 for each subject respectively). The answers varied from approximately 0.5 to 6 seconds, with the average length of an utterance being 1.6 seconds.

The objective of this work is to build automated systems that can discriminate between stress conditions that are categorically different; consequently, it is necessary to devise an annotation of the data that groups categorically similar utterances together for the purposes of modeling the commonalities. The issue of how to label the data is one that deserves particular attention since using different criteria to label the data can (and practically will) lead to different ways to categorically partition the data set, possibly making it more or less challenging to build systems that can discriminate between the selected categories. It is possible, for instance, to assign labels to the data based on the state of the speaker or, conversely, based on some measure which incorporates how listeners perceive the utterances. No single approach is correct, and it is important to bear in mind the application when deciding how to label the data. Cowie (2000) labels these two approaches as cause- and effect-type descriptions to distinguish whether the aim is to capture information about the speaker's state at the time of encoding the speech versus effects on the listener when decoding.

For this work, we have labeled the speech corpus by matching each experimental condition with a distinct category. We feel that this approach is more relevant for stress detection than, for instance, the perceptual similarities and differences, if any, that might exist between the utterances of the corpus. Although we have not conducted any formal perceptual studies on this corpus, the first author has found that perceptual differences between these utterances are not clearly marked; therefore, labeling rules based on experimental conditions may be more suitable for this modeling problem. This approach falls in line with a cause-type description of the data set since it aims to capture the state of the speaker. However, it is important to recognize that even this goal may not be fully achieved since a similar experimental condition may not always translate into the same condition of stress. We have therefore labeled the speech according to the stimulus condition used during the experiment, where each condition was the result of a 2×2 factorial design involving the driving speed (fast or slow) and the frequency with which the driver was prompted to solve the cognitive task (every 4 or 9 seconds).

It is important to bear in mind that we are investigating whether an algorithm can be trained to detect differences in the level of stress, and that in order to do this we have *equated* categorically distinct experimental conditions with categories of stress. The reader should not interpret this to be a statement about the categorical differences (perceptual or otherwise) in the speech. At the onset of this investigation, it was not known whether such differences existed. However, by investigating to what extent machine learning algorithms are able to differentiate between labels established *a priori* on the basis of experimental conditions, we may be able to conclude something about whether these labels actually correspond to different categories of stress, or which of them, if any, is categorically distinct from the rest.

3 Feature Extraction

Nonlinear features of the speech waveform have received much attention in studies of speech under stress; in particular, the Teager Energy Operator (TEO) has been proposed to be robust to noisy environments and useful in stress classification (Zhou et al., 1998; Zhou et al., 1999; Jabloun and Cetin, 1999). Another useful approach for analysis of speech and stress has been subband decomposition or multi-resolution analysis via wavelet transforms (Sarikaya and Gowdy, 1997,1998). Multi-resolution analysis and TEO-based features have also been combined for recognizing speech in the presence of car noise and shown to yield superior rates (Jabloun and Cetin, 1999). In this work we investigate a feature set consisting of variants of features proposed in Jabloun and Cetin (1999) and in Sarikaya and Gowdy (1998) based on the TEO and multi-resolution analysis and apply it to the task of modeling categories of drivers' stress.

The procedure we use is as follows. After sampling the speech signal at 8kHz, multiresolution analysis is applied to the discrete signal $y[n]$ to decompose it into $M = 21$ bands corresponding to the frequency division shown in Figure 1. The decomposition in this implementation is based on repeated iterations of the minimum-phase 8-tap low and high pass filters associated with the orthogonal *Daubechies-4* (Daubechies, 1992). Following the decomposition, the average Teager energy is found for every subband signal according to

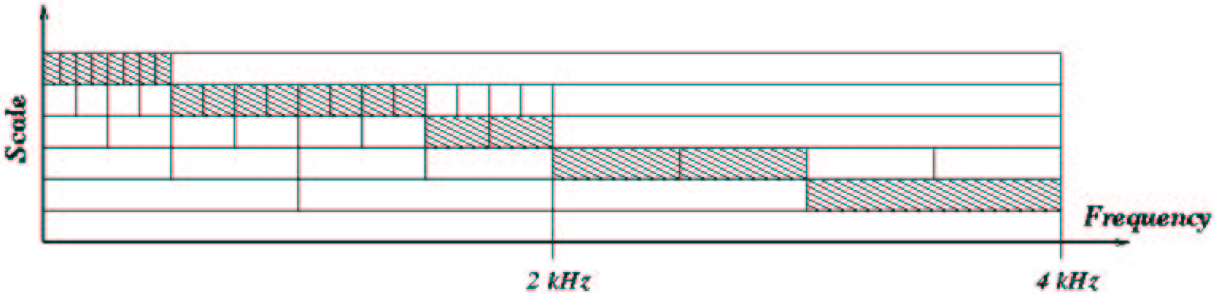

Figure 1: Subband Decomposition

$$e_m = \frac{1}{N_m} \sum_{n=1}^{N_m} |\Psi(y[n])| \quad m = 1, \cdots, M \tag{1}$$

where N_m is the number of time samples in the m^{th} band and $\Psi(\cdot)$ is the discrete Teager energy operator:

$$\Psi(y[n]) = y^2[n] - y[n-1]y[n+1] \tag{2}$$

An inverse DCT transform is then applied to the log of the energy coefficients to obtain the TEO-based "cepstrum coefficients" E_l (Jabloun and Cetin, 1999):

$$E_l = \sum_{m=1}^{M} \log(e_m) \cos\left[\frac{l(m-0.5)\pi}{M}\right] \quad l = 1, \cdots, L \tag{3}$$

The extraction of the cepstral coefficients defined in (3) is applied to the speech waveform at every frame. Define then $\mathbf{E}^{[r]}$ as the $L \times 1$ vector containing the cepstral coefficients from the r^{th} frame: $\mathbf{E}^{[r]} = [E_1^{[r]}, \cdots, E_L^{[r]}]'$. In order to reflect frame-to-frame correlations within an energy subband, the following autocorrelation measure has been proposed (Sarikaya and Gowdy, 1998):

$$ACE_{l,\tau}^{[r]} = \frac{\sum_{n=r}^{r+T} E_l^{[n]} E_l^{[n+\tau]}}{\max_j (\sum_{n=j}^{j+T} E_l^{[n]} E_l^{[n+\tau]})} \quad l = 1, \cdots, L \tag{4}$$

where τ is the lag between frames, T is the number of frames included in the autocorrelation window, and j is an index that spans all correlation coefficients within the same scale along all frames to normalize the autocorrelation. Define the vector containing the logarithm of the L autocorrelation coefficients as $\mathbf{ACE\_L}_{\tau}^{[r]} = [\log ACE_{1,\tau}^{[r]}, \cdots, \log ACE_{L,\tau}^{[r]}]^T$ We define the frame-based feature vector as the set of L cepstral coefficients and the log of the L autocorrelation coefficients:

$$\mathbf{FS}^{[r]} = \begin{bmatrix} \mathbf{E}^{[r]} \\ \mathbf{ACE\_L}_{\tau}^{[r]} \end{bmatrix} \tag{5}$$

Taking the log of (4) is done to avoid modeling a finite support density distribution (which results from the normalization of (4)) with a single or a small number of Gaussians in the learning stage. The number of subbands used in this implementation was $M = 21$, resulting from the 5-level decomposition sketched in Figure 1. The remaining constants were adjusted empirically: $\tau = 1$ and $T = 2$ (to capture pairwise correlations of adjacent samples), and $L = 10$ (to *compress* the original information across 21 bands into 10). The resulting feature vector after appending the correlation coefficients is therefore of dimensionality 20. The frame features are derived from 24 msecs. of speech and are computed every 10 msecs.

4 Modeling Dynamics within the Utterance

4.1 Graphical Models

In this section we treat the dynamic evolution of the utterance features to discriminate between the different categories of driver stress and consider a family of graphical models for time series classification. One of the most extensively studied models in the literature of time series classification is that of a hidden Markov model (HMM). An HMM is often represented as a state transition diagram. Such a representation is suitable for expressing first order transition probabilities; it does not, however, clearly reveal dependencies between variables over time, or clearly encode higher-order Markov structure. Nonetheless, representing an HMM as a dynamical Bayesian net (shown with discrete states s_i and continuous observations x_i in figure 2), allows these statistical dependencies to emerge. This representation also suggests some natural extensions to the structure of the HMM model and aids in the development of general-purpose algorithms that may be used to do learning and inference for a variety of structures. An assumption behind the hidden Markov model, as shown by the dependency diagram of figure 2, is that the observations are independent of each other

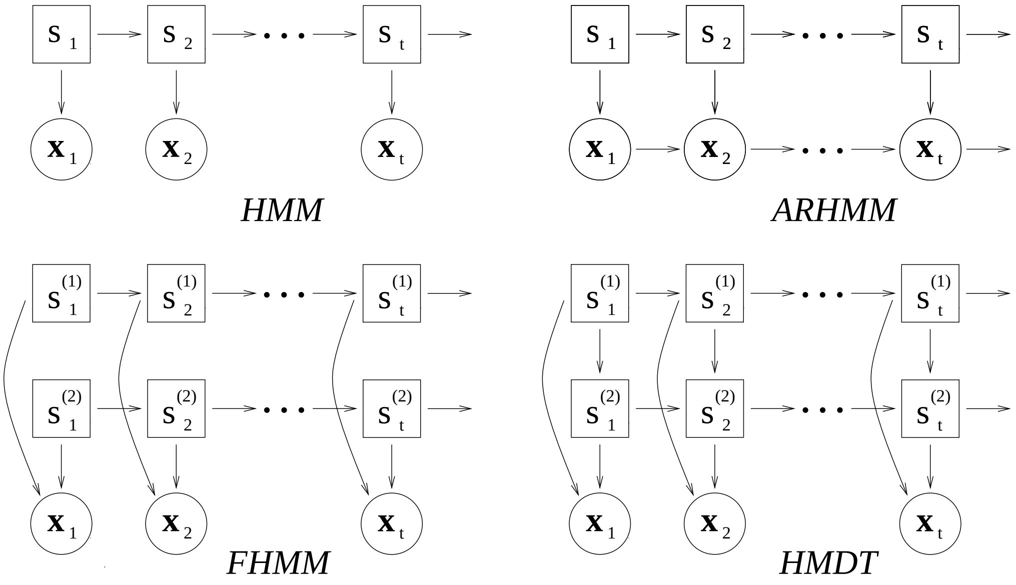

Figure 2: Graphical models compared in this paper

given the hidden state sequence. One may alleviate this limitation by incorporating some dependency on past observations. A simple way to do this is through a first-order recursion on the previous observation. This yields the autoregressive hidden Markov model (ARHMM) (also known as a switching auto-regressive model).

The representational capacity of an HMM is also limited by how closely the number of hidden states approximates the state space of the dynamics. Since the naive way to overcome this limitation –namely, to increase the number of states of the hidden discrete node– yields an increase in the number of parameters to be estimated, distributed state representations which make use of fewer parameters have been proposed. One such structure is the factorial hidden Markov (FHMM) model. In an FHMM the state factors into multiple state variables, each modeled by an independent chain evolving according to the same Markovian dynamics of the basic HMM (Ghahramani and Jordan, 1997).

One can also introduce dependencies between the chains to impose structure while retaining parsimony. For instance, the different state chains can be arranged in a hierarchical structure such that, for any time slice, the state at any level of the hierarchy is dependent on the state at all levels above it. (See figure 2 for a model with 2 chains.) The result – a hidden decision tree evolving over time with Markovian dynamics – is called a hidden Markov decision tree (HMDT) (Jordan, Ghahramani, and Saul, 1997).

The family of graphical models shown in figure 2 has in common a set of unobserved discrete states distributed on a single or multiple chains, and continuous observation nodes. The following formulation of the learning algorithms can be applied to any of the previous structures, as well as to extensions not described here. For instance, a distributed state representation may be combined with an autoregressive hidden Markov model to obtain an autoregressive factorial HMM. We will assume that every discrete node has only discrete parents –that is, the parameters associated with a discrete node consist of a conditional probability table – and that the continuous nodes have a conditional Gaussian distribution. We represent the hidden state as a vector $\mathbf{s}_t = [s_t^{(1)}, \cdots, s_t^{(m)}, \cdots, s_t^{(M)}]'$ to generalize to the case where the hidden state is distributed along several chains, and the observations as the d-dimensional set $\{\mathbf{x}\}_{t=1}^{T}$. In general, a continuous node may have both continuous and discrete parents. Since the kind of dependency on continuous nodes we are interested in is first-order autoregressive, a conditional Gaussian node has distribution

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{s}_t = \mathbf{i}) \sim \mathcal{N}(\mathbf{x}_t; B_{\mathbf{i}} \mathbf{x}_{t-1}, \Sigma_{\mathbf{i}}) \tag{6}$$

where $\mathcal{N}(\mathbf{x}; \mu_x, \Sigma_x)$ is a multivariate Gaussian distribution on the random variable $\mathbf{x}$ with mean vector μ_x and covariance matrix Σ_x. Letting $\mathbf{x}_{t-1} = \mathbf{I}$ (the identity matrix) and $B_{\mathbf{i}} = \mu_{\mathbf{i}}$ in (6), we obtain the distribution on Gaussian nodes with only discrete parents.

We can do learning on these structures by applying the EM algorithm. First, we compute the expected value of the complete data log likelihood given the observations, and holding the current parameters constant (E step). We then maximize the expectation with respect to the parameters to obtain a new estimate (M step). The observation-dependent term of the complete log likelihood is given by

$$\mathcal{L} = E\left[\log \prod_{k}^{K} \prod_{t}^{T_k} \prod_{\mathbf{j}}^{|J|} Pr(\mathbf{x}_t^k | \mathbf{x}_{t-1}^k, \mathbf{s}_t = \mathbf{j}, \{\mathbf{x}_t\}_1^{T_k})^{q_t^{\mathbf{j}}}\right] \tag{7}$$

where $q_t^{\mathbf{j}} \doteq \delta(\mathbf{s}_t = \mathbf{j})$ is an indicator function. Combining (6) and (7), and taking the derivatives of this expectation with respect to the parameters of the distribution, the following estimates are obtained (see Murphy (1998) for derivations):

$$\tilde{B}_{\mathbf{j}} = \left[\sum_{k}^{K} \sum_{t}^{T_k} \gamma_t^k(\mathbf{j}) \mathbf{x}_t^k \mathbf{x}'^k_{t-1}\right] \left[\sum_{k}^{K} \sum_{t}^{T_k} \gamma_t^k(\mathbf{j}) \mathbf{x}_{t-1}^k \mathbf{x}'^k_{t-1}\right]^{-1} \tag{8}$$

$$\tilde{\Sigma}_{\mathbf{j}} = \frac{\sum_k^K \sum_t^{T_k} \gamma_t^k(\mathbf{j}) \mathbf{x}_t^k \mathbf{x}'^k_t}{\sum_k^K \sum_t^{T_k} \gamma_t^k(\mathbf{j})} - \frac{\tilde{B}_{\mathbf{j}} \sum_k^K \sum_t^{T_k} \gamma_t^k(\mathbf{j}) \mathbf{x}_{t-1}^k \mathbf{x}'^k_t}{\sum_k^K \sum_t^{T_k} \gamma_t^k(\mathbf{j})} \tag{9}$$

Equations (8) and (9) are written in terms of the expectations

$$\gamma_t^k(\mathbf{j}) = E[q_t^{\mathbf{j}} | \{\mathbf{x}_t^k\}_1^{T_k}] = Pr(\mathbf{s}_t^k = \mathbf{j} | \{\mathbf{x}_t^k\}_1^{T_k}) \tag{10}$$

Letting $\mathbf{x}_{t-1}^{k} = \mathbf{I}$ and $\tilde{B}_{\mathbf{j}} = \tilde{\mu}_{\mathbf{j}}$ in (8) and (9) yields the estimation equations for the case when the observation nodes are only dependent on the hidden discrete state.

For each discrete node $s_t^{(m)}$, the parameter set consists of a conditional probability table where each entry $\theta_{j,\mathbf{n}}$ is the subtable given by $\Pr\left(s_t^{(m)} = j | \mathrm{pa}(s_t^{(m)}) = \mathbf{n}\right)$, where $\mathrm{pa}(s)$ is the set of parent nodes of state s. B The maximum likelihood estimate of the discrete parameters is then given by

$$\tilde{\theta}_{j,\mathbf{n}} = \frac{\sum_k^K \sum_t^{T_k} \zeta_t^k(j,\mathbf{n})}{\sum_k^K \sum_t^{T_k} \sum_j^{J_m} \zeta_t^k(j,\mathbf{n})} \tag{11}$$

where $\zeta_t^k(j,\mathbf{n}) = \Pr\left({s^k}_t^{(m)} = j, \mathrm{pa}({s^k}_t^{(m)}) = \mathbf{n} | \{\mathbf{x}^k\}_1^{T_k}\right)$.

The EM algorithm consists of iteratively collecting the expected sufficient statistics γ_t and ζ_t in the E step, and updating the parameters of the model according to equations (8)-(11) in the M step. Inference on these graphs (evaluating the marginals above) can be done via the junction tree algorithm (Jensen, 1996). In this scheme, the observations are entered as evidence into the junction tree and propagated. After two full rounds of message passing, the junction tree is consistent (all adjacent cliques agree on the marginal probabilities over their separators), and each clique of the tree contains a joint probability distribution over the clique variables and the entered evidence. The posterior over a variable of interest can then be obtained by marginalization over any clique which contains it. A similar marginalization can be applied to obtain the probability of the observation that is needed in the classification step.

For the implementations reported here, we have modeled the output distributions with unimodal Gaussian densities. The models' free parameters have been chosen as follows: a single HMM with 5 states; a mixture of HMMs with pre-clustering with 5 states on each local model; an FHMM with 2 chains and 2 states per chain; an ARHMM with 1 chain and 3 states per chain; and an HMDT with 2 chains. We used full covariance matrices on the single HMM and on the mixture of HMMs, and diagonal covariance matrices on the remaining models.

4.2 Mixture of Models

In addition to the single architectures just described, we also consider the performance of a mixture model obtained by combining several of the single structures described in the previous section. We have in particular implemented a model which combines several simple HMMs and their individual contributions to classify a time series. Figure 3 shows this model generatively: the n^{th} model is selected with probability α_n (with $\sum_n \alpha_n = 1$), and then a time series is generated according to the parameters λ_n of that model.

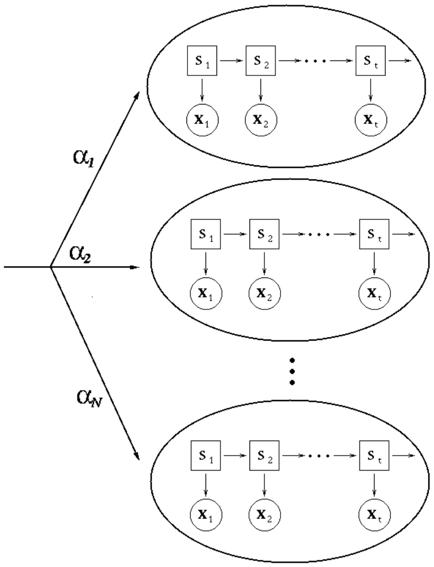

Figure 3: Mixture of HMMs Model

We approach estimating the parameters of such a model for time series classification in two stages. In the first stage, an unsupervised clustering approach is used to discover clusters of the training data in the feature space, where it is assumed the data of each cluster is governed by a single underlying hidden Markov model. In the second stage, the data from each cluster is further used in a supervised approach to create *cluster-dependent class-conditional* models. In this mixture model, therefore, there are two hierarchical levels: at the top level we have a set of N models which partitions the data set, irrespective of the class of the data, into N clusters. At the lower level, we have a set of at most P models (where P is the number of classes) for each of the N clusters.

We estimate the data assignment to the clusters and the parameters of their underlying models by iteratively embedding the HMM training algorithm described in the previous section (which learns the parameters of a cluster given a particular data assignment) within a K-means algorithm (which reassigns time series to clusters according to the probability of membership to each cluster). This algorithm is outlined below:

Given N clusters and a data set consisting of K time series $\mathcal{X} = \{\mathbf{x}^1, \cdots, \mathbf{x}^K\}$, let $\lambda_n^{(l)}$ $(n = 1, \cdots, N)$ be the parameters of the n^{th} HMM at the l^{th} iteration and let $\hat{n}_k^{(l)} = \text{argmax}_n P(\mathbf{x}^k | \lambda_n^{(l)})$ be the cluster that maximizes the probability of the k^{th} time series at the

l^{th} iteration and $\lambda_{\hat{n}_k}^{(l)}$ its parameters.

1. Initialize cluster memberships. Randomly assign time series to clusters to obtain data sets for each cluster $\mathcal{X}_n^{(0)}$. Set $l = 0$.
2. Find initial total log likelihood of the assignment: $P^{(0)} = \sum_k \log P(\mathbf{x}^k | \lambda_{\hat{n}_k}^{(0)})$.
3. For $n = 1, \cdots, N$, apply the parameter re-estimation formulas detailed in the previous section to $\{X\}_n^{(l)}$ to obtain the estimates $\lambda_n^{(l+1)}$.
4. For $k = 1, \cdots, K$ find $\hat{n}_k^{(l+1)} = \text{argmax}_n P(\mathbf{x}^k | \lambda_n^{(l+1)})$ (via the forward-backward or Viterbi algorithms) (Rabiner and Juang, 1993).
5. For $n = 1, \cdots, N$, let $\mathcal{X}_n^{(l+1)} = \{\mathbf{x}^k\}$ for all $\mathbf{x}^k$ whose $\hat{n}_k^{(l+1)} = n$.
6. Find $P^{(l+1)} = \sum_k \log P(\mathbf{x}^k | \lambda_{\hat{n}_k}^{(l+1)})$.
7. If $d(P^{(l+1)}, P^{(l)}) > \epsilon$ (where $d(\cdot, \cdot)$ and ϵ define some convergence criterion), let $l = l+1$ and go to 3; otherwise, stop.

This unsupervised learning procedure is used to identify time series which form clusters in the feature space and is used as a preamble for building cluster dependent supervised learners which exploit the "locality" of data sets in regions of the space. The models in the second stage are therefore trained with only a portion of the categorical data available for each class, namely those time series which are grouped in a common cluster. HMMs have also been used to implement the cluster-dependent class-conditional models at this stage using the same HMM structure and output distribution forms from the unsupervised learning stage. Estimating the parameters of these models can be done following the formalism introduced in the previous section for training graphical models. The results of the unsupervised and supervised learning stages can then be combined for classification using Bayes rule. The posterior probability of a class given an observation is given by summing the cluster- and class-dependent posterior probabilities over the contribution of each of the N clusters:

$$p(\omega | \mathbf{x}_t) = \sum_n^N p(\omega, n | \mathbf{x}_t) = \sum_n^N p(\mathbf{x}_t | \omega, n) p(n | \omega) p(\omega) \tag{12}$$

where the quantity $p(n|\omega)$ can be estimated from the output of the clustering. Assuming equal priors on all classes, and a maximum a posteriori classification scheme, the following decision rule is then obtained:

$$\hat{\omega} = \text{argmax}_l\, p(\omega_l | \mathbf{x}_t) = \text{argmax}_l \sum_n^N p(\mathbf{x}_t | \omega_l, n) p(n | \omega_l) \tag{13}$$

It would be a desirable feature of the unsupervised learning stage to have it yield a partition of the data which is as homogeneous as possible; that is, we would like to obtain a partition which yields clusters with representatives from as few classes as possible. We can diagnose the homogeneity of the clusters by considering the outcome of the unsupervised algorithm in terms of two multinomial variables: the class of the k^{th} data sequence $\mathbf{x}_t^k$ (ω_k) and the cluster to which it is assigned $(c_k \in n = 1 \cdots N)$. The clustering algorithm may then be viewed as yielding a data set $\{\omega_k, c_k\}_{k=1}^K$ to which we want to apply a hypothesis test to determine whether the set of labels and the set of clusters were generated by different multinomial distributions or by the same multinomial. More formally, we would like to know the probability that the sets $\Omega = \{\omega\}_k$ and $C = \{c\}_k$ were generated by the same distribution:

$$\begin{aligned} p(s|\Omega, C) &= \frac{p(\Omega, C|s)p(s)}{p(\Omega, C|s)p(s) + p(\Omega, C|d)p(d)} \\ &= \frac{1}{1 + \frac{p(\Omega,C|d)}{p(\Omega,C|s)}\frac{p(d)}{p(s)}} \end{aligned} \tag{14}$$

where the labels s and d indicate same or different distributions. The main quantity involved in computing (14) is the ratio of evidence of the data sets under different and same distributions $\frac{p(\Omega,C|d)}{p(\Omega,C|s)}$, a quantity which may be written in terms of factorized and joint evidence $\frac{p(\Omega)p(C)}{p(\Omega,C)}$. Let the class ω take on one of J outcomes, and let the cluster c take on one of N outcomes. Define the following partial counts

$$\begin{aligned} K_{j,n} &= \sum_{k=1}^{K} \delta_{w_k,j}\delta_{c_k,n} \\ K_j &= \sum_{n=1}^{N} K_{j,n} \\ K_n &= \sum_{j=1}^{J} K_{j,n} \qquad\qquad j = 1, \cdots, J \quad n = 1, \cdots, N. \end{aligned}$$

It can be shown (Minka, 1999) that, under the assumption of multinomial distributions, the evidence ratio in (14) is given by

$$\frac{p(\Omega)p(C)}{p(\Omega, C)} = \frac{\Gamma(JN)}{\Gamma(K+JN)} \prod_j \frac{\Gamma(K_j+N)}{\Gamma(N)} \prod_n \frac{\Gamma(K_n+J)}{\Gamma(J)} \prod_{j,n} \frac{\Gamma(1)}{\Gamma(1+K_{j,n})} \tag{15}$$

The quantity in (15) also has an interpretation as the mutual information between the variables ω and c (Minka, 1999). Equation (14) may be used to determine whether the clustering procedure has introduced some dependencies between labels and clusters. Furthermore, it may be used together with the clustering algorithm above to select the number of clusters which establishes the largest dependency between variables. For the results reported in this paper, we have considered mixture models with 2 to 6 components, and chosen the number of components which maximizes (14) on the training set.

5 Modeling Features at the Utterance Level

Modeling of linguistic phenomena requires that we choose an adequate time scale to capture relevant details. For speech recognition, a suitable time scale might be one that allows representing phonemes. For the supralinguistic phenomena we are interested in modeling, however, we wish to investigate whether a coarser time scale suffices. The database used in this study consists of short and simple utterances (with presumably simpler structures than those found in unconstrained speech), and hence, global utterance-level features might provide stress discrimination. A simple way to obtain an utterance-level representation of the original dynamic feature set is to use a statistic of each feature time series defined along an utterance (e.g. its sample mean, median, etc.). For the simulations here we have chosen the sample mean of each dynamic feature as the utterance-level feature value. Since the temporal dynamics are now missing, we use static classifiers to discriminate the four categories.

We consider two classification schemes, a support vector machine (SVM) and a neural network (ANN). A SVM implements an approximation to the structural risk minimization principle in which both the empirical error and a bound related to the generalization ability of the classifier are minimized. The SVM fits a hyperplane that achieves maximum margin between two classes; its decision boundary is determined by the discriminant

$$f(\mathbf{x}) = \sum_i y_i \lambda_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{16}$$

where $\mathbf{x}_i$ and $y_i \in \{-1, 1\}$ are the input-output pairs, $K(\mathbf{x}, \mathbf{y}) \doteq \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ is a kernel function which computes inner products, and $\phi(\mathbf{x})$ is a transformation from the input space to a higher dimensional space. In the linearly separable case, $\phi(\mathbf{x}) = \mathbf{x}$. A SVM is generalizable to non linearly separable cases by first applying the mapping $\phi(\cdot)$ to increase dimensionality and then applying a linear classifier in the higher-dimensional space. The parameters of this model are the values λ_i, non-negative constraints that determine the contribution of each data point to the decision surface, and b, an overall bias term. The data points for which $\lambda_i \neq 0$ are the only ones that contribute to (16) and are known as support vectors. Fitting an SVM consists of solving the quadratic program (Osuna et al., 1997):

$$\begin{aligned} \max \quad F(\Lambda) &= \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda \cdot D\Lambda \\ \text{subject to} \quad \Lambda \cdot \mathbf{y} &= 0 \\ \Lambda &\leq C\mathbf{1} \\ \Lambda &\geq \mathbf{0} \end{aligned} \tag{17}$$

where $\Lambda = [\lambda_1 \cdots \lambda_l]'$ and D is a symmetric matrix with elements $D_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and C is a non-negative constant that bounds each λ_i, and which is related to the width of the margin between the classes. Having solved Λ from the equations in (17), the bias term can be found:

$$b = -\frac{1}{2}\sum_i \lambda_i y_i \Big(K(\mathbf{x}_-, \mathbf{x}_i) + K(\mathbf{x}_+, \mathbf{x}_i) \Big) \tag{18}$$

where $\mathbf{x}_-$ and $\mathbf{x}_+$ are any two correctly classified support vectors from classes -1 and $+1$ respectively (Gunn, 1998).

We also consider a two-layer ANN classifier providing a mapping of the form

$$\mathbf{z} = f(\mathbf{x}) = g_2(W_2 g_1(W_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \tag{19}$$

where g_i, W_i and $\mathbf{b}_i$ are the non-linear activation unit, weight matrix and bias vector respectively associated with each layer. We have trained a ANN to minimize the following error criterion

$$E = E_x + E_w = -\sum_i \mathbf{t}_i \cdot \ln(\mathbf{z}_i) + ||\mathbf{w}||^2 \tag{20}$$

where $\mathbf{t}_i$ is a $k \times 1$ vector of zero-one target values encoding the class of the $\mathbf{x}_i$ data point, and $\mathbf{w}$ is a vector containing all the parameters of the network (the entries of W_i and $\mathbf{b}_i$). The first error term (E_x) is the negative cross-entropy between the network outputs and the desired target values. Minimizing this error function is equivalent to maximizing the likelihood of the data set of target values given the input patterns. The second term in (20) (E_w) is a weight decay regularizer that penalizes larger sizes of network parameters (controlling smoothness of the decision surface and regularization ability of the machine) (Bishop, 1995). The weights of the network are updated according to the rule

$$\Delta\mathbf{w} = -\alpha\frac{\delta E}{\delta \mathbf{w}} = -(H + \mu I)^{-1}(\mathbf{g} + \mathbf{w}) \tag{21}$$

where $\mathbf{g} = \sum_i \mathbf{g}^i = \sum_i \frac{\delta E_x^i}{\delta \mathbf{w}}$ is the gradient of the cross-entropy error function with respect to the network weights and $H = \sum_i \mathbf{g}^i (\mathbf{g}^i)^T$ is the outer product approximation to the Hessian matrix. The parameter μ is a momentum parameter chosen adaptively to speed convergence. The derivatives needed to compute (21) are calculated using standard backpropagation.

For the simulations reported here, we have built SVMs with a Gaussian kernel function having width parameter $\sigma = 5$, and two-layer ANNs with 10 and 4 hidden units, and sigmoid and softmax activation units on each layer respectively.

6 Results and Discussion

The speech data of 4 subjects was first divided into a training and testing set comprising approximately 80% and 20% of the data set respectively. The following labels will be used to denote the four categories of data: FF, SF, FS, SS. The first letter denotes whether the data came from a fast (F) or slow (S) driving speed condition; the second indicates the frequency with which the driver was presented with an arithmetic task: every 4 seconds (fast) (F) or every 9 (slow) (S). We applied the models described above to a subject-dependent task (fitting each model to the data of only one subject at a time) as well as to a subject-independent task (pooling the data of all subjects and fitting each model to the global data). The results of the training and testing stage for each one of the subjects as well as the subject-independent results are summarized in Tables 1 through 7.

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overalll
1	72.10	89.47	53.19	76.47	68.25	83.33	75.00	0	75.00	50.00
2	68.42	70.83	70.00	95.24	73.98	41.67	0	23.08	40.00	30.30
3	60.00	25.00	71.43	57.14	56.76	71.43	0	66.67	0	42.31
4	68.42	72.22	43.90	55.56	58.26	66.67	60.00	8.33	42.86	41.67
All	59.74	41.10	34.57	44.16	46.53	48.65	34.09	18.18	15.00	32.52

Table 1: Classification Results (Factorial Hidden Markov Model)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	97.67	94.74	95.74	94.12	96.03	83.33	37.50	50.00	25.00	50.00
2	92.10	91.67	100	100	95.94	66.67	33.33	38.46	0	42.43
3	91.43	75.00	88.57	95.24	88.29	71.43	33.33	55.56	0	46.15
4	86.84	94.44	87.80	100	90.44	66.67	40.00	58.33	0	47.22
All	55.19	55.21	58.02	76.62	59.16	37.84	18.18	27.27	40.00	29.27

Table 2: Classification Results (Single Autoregressive Hidden Markov Model)

Tables 1 through 5 show the results of the five time series classifiers (FHMM, ARHMM,

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	58.14	89.47	57.45	64.71	63.49	100	37.50	20.00	50.00	46.43
2	73.68	62.50	47.50	66.67	61.79	33.33	0	15.38	20.00	21.21
3	71.43	35.00	54.28	66.67	58.56	85.71	0	55.56	0	42.31
4	65.79	77.78	56.10	77.78	66.09	58.33	60.00	25.00	57.14	47.22
All	58.44	49.08	29.63	46.75	48.42	54.05	40.91	13.64	20.00	36.59

Table 3: Classification Results (Hidden Markov Decision Tree)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	95.35	100	94.74	94.12	96.83	100	40.00	25.00	50.00	50.00
2	97.37	90.00	91.67	100	94.31	83.33	38.46	0	40.00	51.52
3	97.14	88.57	60.00	90.48	86.49	85.71	55.56	0	0	42.31
4	100	97.56	100	100	99.13	75.00	66.67	60.00	0	55.56
All	74.68	71.17	38.27	67.53	66.11	67.57	40.91	4.6	30.00	40.65

Table 4: Classification Results (Single Hidden Markov Model)

HMDT, HMM and the mixture of HMM). Tables 6 and 7 summarize the results with a support vector machine (SVM) and a neural network (ANN). In order to be able to assess the performance of each classifier across subjects in the subject-dependent task, the mean value of the overall recognition rates for training and testing sets for each of these classifiers is shown in Table 8. Finally, Tables 9 through 12 shows the confusion matrix obtained for the best performing model (H-HMM) for each subject.

The average overall recognition rates reported in Table 8 show that on the subject-dependent tests the FHMM and HMDT models achieve similar recognition rates on training and testing sets. The HMM and ARHMM also achieve similar recognition rates on both data sets, and both sets of classifiers are outperformed by the M-HMM, which achieves the highest performance of all models considered. The time series classifiers can be ranked according to their performance as follows: M-HMM, HMM, ARHMM, FHMM, HMDT. This ranking is consistent with the performance on both the training and testing sets. The recognition rates of the utterance-level feature set are not significantly different from the recognition rates obtained with the dynamic feature set, except in the case of the M-HMM, where the test set performance is notably better.

When we compare the average performance of the classifiers in the subject-dependent tests (Table 8) with the results obtained in the subject-independent runs (last row of Tables 1 through 7), we can see that the performance degrades in the subject-independent case. The

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	100	97.87	100	100	99.21	83.33	50.00	25.00	0	42.86
2	100	100	100	100	100	100	69.23	0	40.00	69.70
3	97.14	94.29	70.0	85.71	89.19	100	100	83.33	100	96.15
4	100	100	94.44	88.89	97.39	16.67	91.67	0	0	36.11
All	74.68	83.44	81.48	84.42	80.42	48.65	65.91	40.91	35.00	51.22

Table 5: Classification Results (Mixture of HMMs)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	74.42	31.58	57.45	47.06	57.94	66.67	12.50	50.00	75.00	46.43
2	65.79	75.00	42.50	38.10	55.28	75.00	33.33	23.08	80.00	51.51
3	71.43	55.00	77.14	47.62	65.75	71.43	16.67	77.78	0	50.00
4	52.63	66.67	68.29	44.44	59.13	25.00	60.00	58.33	14.28	38.89
All	46.75	22.09	21.00	10.39	28.00	62.16	29.55	18.18	35.00	38.21

Table 6: Classification Results (Support Vector Machine)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	Overall	FF	SF	FS	SS	Overall
1	86.05	73.68	91.49	88.23	86.51	33.33	50.00	20.00	75.00	39.28
2	86.84	91.67	90.00	61.90	84.55	50.00	33.33	53.85	40.00	48.48
3	85.71	65.00	94.26	66.67	81.08	100	33.33	77.78	0	61.53
4	76.32	55.56	90.24	61.11	75.65	58.33	60.00	66.67	14.28	52.77
All	56.50	63.80	0	0	40.21	59.46	61.36	0	0	39.84

Table 7: Classification Results (Neural Network)

Models	Training (%)	Testing (%)
FHMM	64.31	41.07
ARHMM	92.67	46.45
HMDT	62.48	39.29
HMM	94.78	49.85
M-HMM	96.44	61.20
SVM	59.52	46.70
ANN	81.94	50.57

Table 8: Mean Recognition Rates for all Classifiers in the Subject-Dependent Task

mixture model proposed in this paper, however, still manages to outperform the competing models in the subject-independent case. It can be argued that some of the robustness of the mixture model in capturing variability *within* a subject carries over when handling the variability *across* subjects since the unsupervised clustering phase tends to divide the data into more homogeneous subsets irrespective of the speaker.

	Training				Testing			
	FF	SF	FS	SS	FF	SF	FS	SS
FF	43	0	0	0	5	1	0	0
SF	0	46	0	1	4	5	1	0
FS	0	0	19	0	1	5	2	0
SS	0	0	0	17	1	3	0	0

Table 9: Confusion Matrices on Training and Testing Sets for Subject 1 (M-HMM model).

	Training				Testing			
	FF	SF	FS	SS	FF	SF	FS	SS
FF	38	0	0	0	12	0	0	0
SF	0	40	0	0	1	9	3	0
FS	0	0	24	0	2	1	0	0
SS	0	0	0	21	3	0	0	2

Table 10: Confusion Matrices on Training and Testing Sets for Subject 2 (M-HMM model).

Since we are modeling four different categories, the expected overall performance of a classifier which labels the data at random is 25%. All classifiers exceed this overall performance although they fall short of achieving a near-perfect recognition rate. However, it is important that we bear in mind that a perfect recognition rate may be a too optimistic figure. Humans, for instance, consistently achieve recognition rates below 100% on perceptual experiments on decoding affect from speech. Bezooijen (1984) reports recognition rates on a set of five discrete affective states (fear, disgust, joy, sadness and anger) that range from 49% to 74% whereas Pittam and Scherer (1993) report rates on the same set of affective states ranging from 28% to 72%. More recent work by Polzin (2000) reports human recognition figures on a set of four affective states labeled as fear, happiness, sadness and anger ranging from 58.5% to 80.1% with an overall rate of 69%.

For systems that aim to recognize the perceptual labels which humans "hear" (a system using an annotation convention driven by the effect of the utterances on the listeners, as we discussed in section 2), the human performance figures offer a reasonable yardstick by which we should evaluate the performance of an automated system. However, for systems aiming to recognize labels governed by the state of the speaker (cause-type of annotation), establishing

	Training				Testing			
	FF	SF	FS	SS	FF	SF	FS	SS
FF	34	0	1	0	7	0	0	0
SF	1	33	0	1	0	9	0	0
FS	2	4	14	0	1	0	5	0
SS	1	1	1	18	0	0	0	4

Table 11: Confusion Matrices on Training and Testing Sets for Subject 3 (M-HMM model).

	Training				Testing			
	FF	SF	FS	SS	FF	SF	FS	SS
FF	38	0	0	0	2	10	0	0
SF	0	41	0	0	1	11	0	0
FS	1	0	17	0	2	3	0	0
SS	1	1	0	16	0	7	0	0

Table 12: Confusion Matrices on Training and Testing Sets for Subject 4 (M-HMM model).

a benchmark is less straightforward. Comparison with other studies that use a similar labeling approach to build automatic recognizers may be useful to obtain an impressionistic idea of what is accomplishable. However, as the context in which the utterances are produced often varies significantly across these studies, it is important that we carry such comparisons with due care. Even affective states that are labeled with the same descriptor may correspond to very different realities about the internal state of the speaker. McGilloway et al. (2000) have carried out a study in which they used emotive texts to induce affect in speech. This study can therefore be considered similar in so far as the labels capture information about the state of the speaker. They report recognition rates ranging from 50% to 64% for a set of five states (fear, happiness, neutral, sadness and anger). Bearing the above mentioned caveats in mind, we can see the performance that the M-HMM achieves on some subjects as being competitive with some of the results published elsewhere in the literature.

There is variability, however, on how these systems perform across subjects. This result may be due to the fact that subjects with the lower recognition rates do not show sufficient variability across the categories we are modeling, but it may also be attributed to a limitation of the models or the features in capturing the variability that the subjects may have exhibited. This is an issue that would require further research in order to draw more rigorous conclusions. It is also important to note the variability of the classifiers in modeling each of the categories considered *independently*. Whereas all the models provide an adequate fit to the FF category, each of them fails to consistently predict above random the remaining categories for all subjects (see Tables 1-7). This may be due to one or more of several reasons:

(i) the inherent modeling capacity of the models considered, (ii) an underoptimized local solution found during training, (iii) the discriminative capacity of the features for the different categories, or (iv) the inherent noise in the ground truth of the categories of driver's stress due to how accurately the experimental procedure was able to effectively induce the assigned labels. Since the FF category is the most "extreme" in terms of driving speed and cognitive load on the driver, it is tempting to assume that the better performance on this label may be related to how reliably the driver became stressed in these portions of the experiment. We have also included for completeness confusion matrices that show the kinds of missclassifications that the best performer (mixture model) tends to produce (Tables 9-12). There is no consistent pattern that emerges from inspecting these confusion matrices: no category is consistently mistaken for another one when the results are evaluated across subjects. We have dealt with subject dependency by fitting a different system to each speaker. One open question worth investigating is whether this dependency can be reduced without having to go to the extreme of fitting a different set of parameters to each speaker. An alternative solution might be finding *prototypes* of speakers in the space of speaker variability and fit a model to each prototype rather than to each speaker. We are advancing the hypothesis that the mixture model might offer a good point of departure for this kind of modeling (as well as modeling any categories involving variability across speakers) because of how it proceeds in dividing the feature space into similar clusters. A dataset with a larger number of speakers would be needed, however, in order to evaluate this hypothesis.

7 Conclusions

In this paper we have investigated the use of features based on subband decompositions and the TEO for classification of stress categories in speech produced in the context of driving at variable speeds while engaged on mental tasks of variable cognitive load for a set of 4 subjects. We investigated the performance of several classifiers on two representations of the speech waveforms: using a feature set representing intra-utterance dynamics and a sparser set consisting of more global utterance-level features. The best performance was obtained by using the dynamic feature set and by exploiting local models and then combining them in a weighted classification scheme. All classifiers produced recognition rates above random for all subjects, but, with the exception of the fast-fast category, showed variability in consistently predicting each of the remaining stress conditions.

Acknowledgments

The authors would like to thank Nissan's CBR Lab and Elias Vyzas for their help with data collection and Thomas Minka for valuable technical discussions and for suggesting the significance test.

References

Bezooijen, R. v. (1984). Characteristics and Recognizability of Vocal Expressions of Emotion. Holland: Foris Publications.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

Cowie, R. (2000). Describing the emotional states expressed in speech. In ISCA Workshop on Speech and Emotion. (pp. 11-18). Newcastle, Northern Ireland.

Daubechies, I. (1992). Ten Lectures on Wavelets. SIAM.

Ghahramani, Z., and Jordan, M. I. (1997). Factorial hidden markov models. Machine Learning, 29, 245-275.

Gunn, S. (1998). Support vector machines for classification and regression (Tech. Rep.). Image, Speech and Intelligent Systems Group. University of Southampton.

Hansen, J., Bou-Ghazale, S. E., Sarikaya, R., and Pellom, B. (1998). Getting started with the SUSAS: Speech Under Simulated and Actual Stress database (Tech. Rep.). Robust Speech Proc. Laboratory. Dept. of Electrical Engineering. Duke University.

Hansen, J. H. L., and Womack, B. D. (1996). Feature analysis and neural network-based classification of speech under stress. IEEE Transactions on Speech and Audio Processing., IV(4), 307-313.

Jabloun, F., and Cetin, A. E. (1999). The Teager energy based feature parameters for robust speech recognition in car noise. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. I, pp. 273-276).

Jensen, F. V. (1996). An Introduction to Bayesian Networks. Springer.

Jordan, M. I., Ghahramani, Z., and Saul, L. K. (1997). Hidden Markov decision trees. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), Advances in Neural Information Processing Systems (Vol. 9, pp. 501-507). Cambridge, Massachusetts: MIT Press.

McGilloway, S., Cowie, R., Douglas-Cowie, E., Gielen, S., Westerdijk, M., and Stroeve, S. (2000). Automatic recognition of emotion from voice: A rough benchmark. In ISCA Workshop on Speech and Emotion (pp. 207-212). Newcastle, Northern Ireland.

Minka, T. P. (1999). Bayesian inference of a multinomial distribution. (http://www.media.mit.edu/~tpminka/
papers/tutorial.html)

Murphy, K. (1998). Fitting a constrained conditional Gaussian (Tech. Rep.). Dept. of Computer Science. U.C. Berkeley.

Murray, I. R., Baber, C., and South, A. J. (1996). Towards a definition and working model of stress and its effects on speech. Speech Communication, 20, 1-12.

Osuna, E. E., Freund, R., and Girosi, F. (1997). Support vector machines: Training and applications (Tech. Rep. No. A.I. Memo 1602/C.B.C.L. Paper 144). MIT.

Pittam, J., and Scherer, K. (1993). Vocal expression and communication of emotion. In M. Lewis and J. M. Haviland (Eds.), Handbook of Emotions (pp. 185-197). New York: Guilford Press.

Polzin, T. (2000). Detecting verbal and non-verbal cues in the communication of emotions. Unpublished doctoral dissertation, School of Computer Science. Carnegie Mellon University.

Rabiner, L., and Juang, B.-H. (1993). Fundamentals of Speech Recognition. Prentice Hall.

Sarikaya, R., and Gowdy, J. N. (1997). Wavelet based analysis of speech under stress. In Southeastcon '97. Engineering New Century. Proceedings IEEE. (pp. 92-96).

Sarikaya, R., and Gowdy, J. N. (1998). Subband based classification of speech under stress. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. I, pp. 569-572).

Steeneken, H. J. M., and Hansen, J. H. L. (1999). Speech under stress conditions: Overview of the effect on speech production and of system performance. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. IV, pp. 2079-2082).

Zhou, G., Hansen, J., and Kaiser, J. (1998). Classification of speech under stress based on features derived from the nonlinear Teager energy operator. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. I, pp. 549-552).

Zhou, G., Hansen, J. H. L., and Kaiser, J. F. (1999). Methods for stress classification: Nonlinear TEO and linear speech based features. In IEEE International Conference on Acoustics, Speech, and Signal Processing. (Vol. IV, pp. 2087-2090).