M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 528
Submitted to the ISCA Workshop on Speech and Emotions, Belfast, 2000

MODELING DRIVERS' SPEECH UNDER STRESS

Raul Fernandez and Rosalind W. Picard

MIT Media Laboratory
20 Ames St., Cambridge, MA 02139
{galt,picard}media.mit.edu

Abstract

In this paper 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 of the data. This leads to two sets of modeling techniques. In the first approach, we model the dynamics of the feature set *within* the utterance with a family of dynamic classifiers. In the second approach, we model the mean value of the features *across* the utterance with a family of static classifiers. We report and compare classification performances on the sparser and full dynamic representations for a set of four subjects.

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 have targeted perceptual (e.g. Lombard effect), psychological (e.g. timed tasks), as well as physical stressors (e.g. roller-coaster rides, high G forces) [1]. In this work we are interested in modeling speech in the context of driving under varying conditions of cognitive load which are hypothesized to induce a level of stress on the driver. The results of this research may be not only relevant to building recognition systems that are more robust in the context described, but also applicable to and inspired by applications that may 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.

This work was supported by the MIT Media Lab Digital Life Consortium.

The recent literature discussing the effects of stress on speech applies the label of *stress* to different acoustic phenomena. Following the taxonomy proposed by Murray et al. [2], we are investigating the effect on speech of what the authors call "third-order stressors," that is, the effect of external stimuli as well as underlying affective conditions.

2 SPEECH DATABASE

The speech data was collected in an experiment in a driving simulator at the Nissan's Cambridge Research Lab. Subjects were asked to complete a series of rounds while engaged on 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 at which the driver had to solve the math questions. Subjects drove at 60 m.p.h. in the low speed condition and at 120 m.p.h. in the high speed condition (the perceptual speed in the simulator is approximately half). When a subject complained of motion sickness in the high speed condition, the speed was reduced to 100 m.p.h. The frequency at which the driver was prompted for an answer was once every 9 seconds in the slow condition, and once every 4 seconds in the fast condition. The driver's answers were captured by a head-mounted microphone and recorded in VHS format.

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 [3],[4], [5]. Another useful approach for analysis of speech and stress has been subband decomposition or multi-resolution analysis via wavelet transforms [6],[7]. 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 [5]. In this work we investigate a feature set consisting of variants of features proposed in [5] and [7] based on the TEO and multi-resolution analysis and apply it to the task of modeling categories of drivers' stress.

After sampling the speech signal at 8kHz, multiresolution analysis is applied to the discrete signal $x[n]$ to decompose it into $M = 21$ bands corresponding to the frequency division shown in Figure 1. The decomposi-

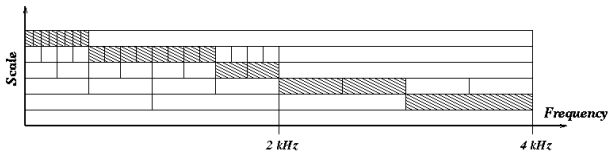

Figure 1: Subband Decomposition

tion 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* [8]. Following the decomposition, the average Teager energy is found for every subband signal according to

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

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

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

An inverse DCT transform is then applied to the log of the energy coefficients to obtain the TEO-based "cepstrum coefficients" E_l [5]:

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

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: $\mathbf{E}^{[r]} = \left[E_1^{[r]}, \cdots, E_L^{[r]}\right]^T$. In order to reflect frame-to-frame correlations within an energy subband, the following autocorrelation measure has been proposed [7]:

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

where τ is the lag between frames, T is the number of frames included in the autocorrelation window, and j is an index which 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 $\mathbf{ACE\_L}_\tau^{[r]} = \left[\log ACE_{1,\tau}^{[r]}, \cdots, \log ACE_{L,\tau}^{[r]}\right]^T$ 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 values of the constants for this implementation are $M = 21$, $\tau = 1$, $T = 2$, and $L = 10$ (resulting in a feature vector of dimensionality 20). The frame features are derived from 24 msecs. of speech and are computed every 10 msecs.

4 MODELING

4.1 Dynamics within the Utterance

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 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. Representing an HMM as a dynamical Bayesian net (figure 2), however, 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 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

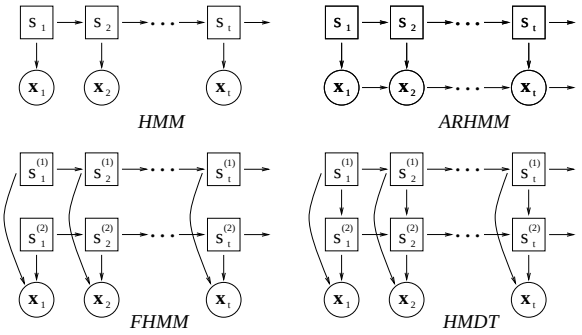

Figure 2: Graphical Models

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.

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).

In addition to the single architectures just described, we also consider the performance of a composite model obtained by fitting several single HMMs to clusters of the data set and combining each model's classification of a time series. HMM parameters and cluster memberships are iteratively estimated by embedding the HMM training algorithm (which learns the parameters of a cluster given its data assignment) within a K-means algorithm (which assigns time series to clusters according to the probability of membership to each cluster). Since the K-means algorithm requires a pre-specified number of clusters K, we vary K from 2 to 6 clusters, apply the procedure described above, and retain the clustering which maximizes a homogeneity test between the classes and the clusters Intuitively we keep the value of K which yields the *purest* clusters, with most of the cluster members being of the same class. (See [9] for details on the homogeneity test.)

4.1.1 Learning and Inference.

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 (CPT)– and that the continuous nodes have a conditional Gaussian distribution. We represent the hidden state as a vector $\mathbf{s}_t = [s_t^{(1)}, \cdots, s_t^{(M)}]^T$ 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

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

Letting $\mathbf{x}_{t-1} = 1$ 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), and 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

$$\mathcal{L} = E\left[\log \prod_t \prod_{\mathbf{i}} Pr(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{s}_t = \mathbf{i}, \{\mathbf{x}\}_1^T)^{q_t^{\mathbf{i}}}\right] \qquad (7)$$

where $q_t^{\mathbf{i}} \doteq \delta(\mathbf{s}_t = \mathbf{i})$ 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 [10] for derivations):

$$\tilde{B}_{\mathbf{i}} = \left[\sum_t \gamma_t(\mathbf{i})\mathbf{x}_t\mathbf{x}'_{t-1}\right]\left[\sum_t \gamma_t(\mathbf{i})\mathbf{x}_{t-1}\mathbf{x}'_{t-1}\right]^{-1} \qquad (8)$$

$$\tilde{\Sigma}_{\mathbf{i}} = \frac{\sum_t \gamma_t(\mathbf{i})\mathbf{x}_t\mathbf{x}'_t}{\sum_t \gamma_t(\mathbf{i})} - \frac{\tilde{B}_{\mathbf{i}}\sum_t \gamma_t(\mathbf{i})\mathbf{x}_{t-1}\mathbf{x}'_t}{\sum_t \gamma_t(\mathbf{i})} \qquad (9)$$

Equations (8) and (9) are written in terms of the expectations $\gamma_t(\mathbf{i}) = E[q_t^{\mathbf{i}}|\{\mathbf{x}\}_1^T] = Pr(\mathbf{s}_t = \mathbf{i}|\{\mathbf{x}\}_1^T)$. Letting

$\mathbf{x}_{t-1} = 1$ and $\tilde{B}_{\mathbf{i}} = \tilde{\mu}_{\mathbf{i}}$ 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 CPT where each entry $\theta_{j,\mathbf{i}}$ is the subtable given by $\Pr(s_t^{(m)} = j | \mathrm{pa}(s^{(m)}) = \mathbf{i})$. The maximum likelihood estimate of the discrete parameters is then given by

$$\tilde{\theta}_{j,\mathbf{i}} = \frac{\sum_t \zeta_t(j,\mathbf{i})}{\sum_t \sum_j \zeta_t(j,\mathbf{i})} \tag{10}$$

where $\zeta_t(j,\mathbf{i}) = \Pr\Big(s_t^{(m)} = j, \mathrm{pa}\big(s_t^{(m)}\big) = \mathbf{i} \big| \{\mathbf{x}\}_1^T\Big)$.

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)-(10) in the M step. Inference on these graphs (evaluating the marginals above) can be done via the junction tree algorithm. 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 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

$$f(\mathbf{x}) = \sum_i y_i \lambda_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{11}$$

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 (11) and are known as support vectors. Fitting a SVM consists of solving the quadratic program [11]:

$$\begin{aligned} \max \quad F(\Lambda) &= \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda \cdot D\Lambda \\ \text{subject to} \quad \Lambda \cdot \mathbf{y} &= 0 \\ \Lambda &\leq C\mathbf{1} \\ \Lambda &\geq \mathbf{0} \end{aligned} \tag{12}$$

where $\Lambda = [\lambda_1 \cdots \lambda_l]^T$ and D is a symmetric matrix with elements $D_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. 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 (12), the bias term can be found:

$$b = -\frac{1}{2} \sum_i \lambda_i y_i \Big(K(\mathbf{x}_-, \mathbf{x}_i) + K(\mathbf{x}_+, \mathbf{x}_i) \Big) \tag{13}$$

where $\mathbf{x}_-$ and $\mathbf{x}_+$ are any two correctly classified support vectors from classes -1 and $+1$ respectively [12].

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

$$\mathbf{z} = f(\mathbf{x}) = g_2(W_2 g_1(W_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \qquad (14)$$

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

$$E = E_x + E_w = -\sum_i \mathbf{t}_i \cdot \ln(\mathbf{z}_i) + ||\mathbf{w}||^2 \qquad (15)$$

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 (15) (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) [13]. The weights of the network are updated according to the rule

$$\Delta \mathbf{w} = -\alpha \frac{\delta E}{\delta \mathbf{w}} = -(H + \mu I)^{-1}(\mathbf{g} + \mathbf{w}) \qquad (16)$$

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 (16) 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.

5 RESULTS AND DISCUSSIONS

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) speed condition; the second indicates the frequency with which the driver was engaged in solving a task: every 4 seconds (fast) (F) or every 9 (slow) (S). The results of the training and testing stage for each one of the subjects for the models previously discussed appear in Tables 2 through 8.

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 1: Mean Recognition Rates for all Classifiers

Tables 2 through 6 show the results of the five time series classifiers (FHMM, ARHMM, HMDT, HMM and the mixture of HMM). Tables 7 and 8 summarize the results with a support vector machine (SVM) and a neural network (ANN). The mean value of the overall recognition rates for training and testing sets for each of these classifiers is shown in Table 1.

The average overall recognition rates reported in Table 1 show that 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.

It is also important to note the variability of these classifiers in modeling each of the categories considered. 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 2-8). 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

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 2: Classification Results (Factorial Hidden Markov Model)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 3: Classification Results (Single Autoregressive Hidden Markov Model)

on this label may be related to how reliably the driver became stressed in these portions of the experiment.

6 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.

References

[1] Herman J.M. Steeneken and John H.L. Hansen. Speech under stress conditions: Overview of the effect on speech production and of system performance. In *Proceedings ICASSP '99*, volume 4, pages 2079–2082, 1999.

[2] I.R. Murray, C. Baber, and A.J. South. Towards a definition and working model of stress and its effects on speech. *Speech Communication*, 20:1–12, November 1996.

[3] Guojun Zhou, John H.L. Hansen, and James Kaiser. Classification of speech under stress based on features derived from the nonlinear Teager energy operator. In *Proceedings ICASSP '98*, volume 1, pages 549–552, 1998.

[4] Guojun Zhou, John H.L. Hansen, and James F. Kaiser. Methods for stress classification: Nonlinear TEO and linear speech based features. In *Proceedings ICASSP '99*, volume 4, pages 2087–2090, 1999.

[5] Firas Jabloun and A. Enis Çetin. The Teager energy based feature parameters for robust speech recognition incar noise. In *Proceedings ICASSP '99*, volume 1, pages 273–276, 1999.

[6] Ruhi Sarikaya and John N. Gowdy. Wavelet based analysis of speech under stress. In *Southeastcon '97. Engineering new New Century. Proceedings IEEE*, pages 92–96, 1997.

[7] Ruhi Sarikaya and John N. Gowdy. Subband based classification of speech under stress. In *Proceedings ICASSP '98*, volume 1, 1998.

[8] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[9] Raul Fernandez and Rosalind W. Picard. Analysis and classification of stress categories from drivers'

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 4: Classification Results (Hidden Markov Decision Tree)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 5: Classification Results (Single Hidden Markov Model)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 6: Classification Results (Mixture of HMMs)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 7: Classification Results (Support Vector Machine)

Subject	Training Rec. Rates (%)					Testing Rec. Rates (%)				
	FF	SF	FS	SS	All	FF	SF	FS	SS	All
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

Table 8: Classification Results (Neural Network)

speech. Technical Report 513, MIT Media Lab, 1999.

[10] K. Murphy. Fitting a constrained conditional gaussian. Technical report, U.C. Berkeley, 1998.

[11] R. Freund E.E. Osuna and F. Girosi. Support vector machines: Training and applications. Technical Report A.I. Memo 1602/C.B.C.L. Paper 144, MIT, 1997.

[12] S. Gunn. Support vector machines for classification and regression. Technical report, Image, Speech and Intelligent Systems Group. University of Southampton, 1998.

[13] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.