MIT Media Laboratory Technical Report 446
Also appears as MIT thesis for the degree of
Master of Science in Electrical Engineering and Computer Science
Supervised by Rosalind W. Picard

Stochastic Modeling of Physiological Signals with Hidden Markov Models: A Step Toward Frustration Detection in Human-Computer Interfaces.

Raul Fernandez
Room E15-394, The Media Laboratory
Massachusetts Institute of Technology
20 Ames St., Cambridge, MA 02139
galt@media.mit.edu

Abstract

Affective Computing is a newly emerging field which has been defined as "computing that relates to, arises from, or deliberately influences emotions." Many applications in affective computing research rely on, or can greatly benefit from, information regarding affective states of computer users. The sensing and recognition of human affect, therefore, is one of the most important research areas to receive much attention in the area of affective computing. In this work, inspired by the particular application of human-machine interaction and the potential use that human-computer interfaces can make of knowledge regarding the affective state of a user, we investigate the problem of sensing and recognizing typical affective experiences that arise in this setting. In particular, through the design of experimental conditions for data gathering, we approach the problem of detecting "frustration" in human computer interfaces. By first sensing human biophysiological correlates of internal affective states, we proceed to stochastically model the biological time series with Hidden Markov Models to obtain user-dependent recognition systems that learn affective patterns from a set of training data. Labeling criteria to classify the data are discussed, and generalization of the results to a set of unobserved data is evaluated. Final recognition results are reported under two conditions, for the entire data set, and only for those subjects with sufficient experimental data. Under the first criterion, recognition rates greater than random are obtained for $\frac{2}{3}$ of the subjects whereas under the second criterion, significant recognition rates are reported for $\frac{7}{8}$ of the subjects.

This work was funded by IBM, BT, HP, and the MIT Media Lab TTT Consortium

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the advent of the digital era, computers have rapidly grown to exhibit great computational capabilities. Their great potential in assisting human problem solving has turned them into a ubiquitous aspect of modern life. Given the currently accessible "state-of-the-art" in computing and communication technologies, human-machine interaction is a phenomenon that absorbs much of our time on a daily basis. However, in spite of all these advances, there is an area in which computers still fail to excel: the expression and recognition of affect in the user population with which they interact.

Computer's behavior is currently and foremost the result of taut rules which pay little attention to or completely bypass the user's affective state. Computers are currently able to properly sense, decode, and react to certain aspects of human behavior which are easily quantifiable (i.e. making a syntax error), but are considerably less efficient when the domain of expression is human affective expression (i.e. a user showing signs of frustration while trying to use a software package). There has been considerable interest lately in bringing forth the role of affect. This interest is largely indebted to recent neurological studies which have revised and reinterpreted the role of emotions in human cognition and transformed emotions research from a dormant to an active research area in the psychology field. The emergence of Affective Computing as a research field also follows this newly found interest in understanding the function of affect.

Affective computing is "computing that relates to, arises from, or deliberately influences emotions" [1]. This work addresses one aspect of affective computing research, namely the recognition of human affect for applications arising naturally in a computer-human interaction. The organization of this thesis is as follows: in the remaining of this chapter, we present an overview of research areas which have provided some of the inspiration for this project, as well as an overview of related research work. In Chapter 2, we offer a review of the theoretical framework used in this project for the modeling of affective signals. In Chapter 3, several implementation issues are discussed. Results from this work are detailed in Chapter 4, and in the last chapter we conclude by suggesting future research steps.

1.1 Background and Related Research

A great inspiration for the development of affective computing research and the resurgence of interest in the area of emotions was provided by the work of Antonio Damasio. Damasio's research [2] was one of the first to shed a new light on the role of affect by demistifying centuries-old conceptions of the marginal and often negative functions of emotions in human rational thinking. Damasio's research challenged the Cartesian view of the "body-mind" dichotomy, and helped establish the notion that emotions are indeed a strong component of rational decision making processes. Another substantial contribution to emotion research has been provided by the recent work of the neuropsychologist Joseph LeDoux whose work has helped elucidate the neurological origins of emotions and their involvement in perception [3].

Much work has originated on how affect can be transmitted. Our approach to this work is based on principles of human psychophysiology, a subject with a rich history of research. There are numerous (and often contradictory) claims in the psychophysiology literature that internal affective states can be mapped onto physiological states [4], [5], [6]. The nature of these relations is an ongoing topic of research, and some authors have suggested treating this as a pattern recognition problem, an approach that is often absent from the psychology literature [7]. Historically, the sensing of human physiological responses to draw inferences about the internal state of an individual has spawned several applications including deception detection (polygraph) and alertness monitoring. Recently there has been interest in applying more sophisticated computational models to some of these research areas by importing principles from such areas as feedback systems [8], and neural networks [9].

In the context of computers, the notion of intelligent artificial systems capable of some sort of emotional processing is not entirely new. Sloman argued in 1981 for the need of systems with such capabilities [10]. Recent work in the computer vision field has paid attention to the recognition and classification of facial expressions [11], and there has been considerable interest in the speech processing community in building systems that incorporate affective expression into synthesized speech [12].

1.2 Problem Statement

There exists a variety of challenging open research problems in the area of "affective computing." At the core of this lies the issue of emotion detection and classification. There is an extensive and controversial literature on emotion theory which attempts to establish models of emotions. Whereas these models do not always agree in describing what emotions are, many authors resort to a simplified model that allows them to establish some quantifiable dimensions along which we can describe affective states. One such model places emotions on a two dimensional map having *arousal* and *valence* (the positive or negative character of an affective state) [13] as its two independent axes. We propose a basic approach to assess the affective state of a user by designing computational models that are capable of detecting and evaluating these sub-components of emotions. In particular, the aim of this project is to create a system that learns the typical responses of a user during a high arousal situation, in particular the kind of frustration that arises while interacting with a computer. The output of this system can then be coupled with that of a system which evaluates valence to formulate a more refined prediction of an affective state. The emphasis of this work is on affect detection and classification. It should be noted, however, that this work is motivated by and has implications for other functionalities of a complete affective system. For instance, the ability to evaluate the affective state of a user can help the interface designer build adaptive affective interfaces.

A desirable property of this work is that it be applicable to a real life situation in which we can envision an affective computer at work. Motivated by the increasing accessibility of computers to a (not always computer literate) population, we have decided to work in the domain of human-machine interaction, and specifically look at situations in which the quality of the interaction is degraded by the computer's failure to adapt to the needs of the user. The objective is to confront the user with a scenario in which the computer deliberately and randomly fails in some functionality which frustrates the user. While the user interacts with the computer, a series of physiological signals are continuously collected and sampled for further analysis. It is further desired to implement algorithms that can be executed in real time.

1.3 Experiment and Methodology

One of the major obstacles in implementing a system that acts on affective signals resides in collecting suitable data for developing a computational model for the system. The data used should, to the best of our knowledge, reflect the affective state we are interested in modeling. This often turns out to be a difficult task since affective states are difficult to induce, and this is further constrained by the fact that subjects are often brought into a laboratory setting. The collection of real life data under similar conditions for all subjects would be impractical and very time consuming. Hence, the collection of affective signals has to be guided by a rigorous experiment in which the subject has no knowledge of the purpose [1]. A controlled experiment was therefore developed to gather data. The methodology is described in detail in Appendix A. The signals collected were galvanic skin response (GSR) and blood volume pressure (BVP). Chapter 3 discusses some of the issues relating to the collected signals, as well as the important issue of establishing a ground truth for classification.

[1] It is arguable that a user aware of the process of bringing him/her to a high arousal state might show a qualitatively different response. Alternatively, a user may be asked to bring him/herself to a high arousal state (by, for instance, imagining a high arousal situation). Again, this can affect the response. In fact, self-expressed affect is an area of current research at this lab.

Chapter 2

Modeling

The main objective of this section is to develop models and techniques which we can apply in real time to track physiological signals and make inferences about the level of arousal of a subject. We envision this project being a useful building block that can be integrated into a computer that uses this information to adapt itself to the needs of the user. This more ambitious idea goes beyond the present scope of this thesis but is a future research topic in this area.

Human physiology behaves like a complex dynamical system in which several factors, both external and internal, shape the outcome. In approximating such a system, we are interested in modeling its dynamical nature and, given that knowledge of all the independent variables that affect the system is limited, we want to approach the problem in a stochastic framework that will help us model the uncertainty and variability that arise over time. A class of models that has received much attention in the research community over past years to model complex dynamic phenomena of a stochastic nature is the class of Hidden Markov Models (HMM). HMMs have been widely used for modeling speech and gesture, and are currently an important building block of speech recognition systems. Motivated by their flexibility in modeling a wide class of problems, we decided to study the feasibility of using HMMs to model physiological patterns that are believed to correlate with different affective states.

2.1 Hidden Markov Models

2.1.1 Markov Processes

Stochastic processes "with memory" are processes in which the present value of the process depends on the recent history undergone by the process. Let us consider a discrete process generated by a random variable s which at time t takes a value on a finite set $S = \{S_1, S_2, \cdots, S_N\}$. Such a discrete stochastic process is said to be j^{th} order Markov if it satisfies the Markov property; that is, the conditional probability of the process given all past and present values depends only on the j most recent values:

$$P(s_t|s_{t-1}, s_{t-2}, \cdots, s_{t-j}, s_{t-j-1}, \cdots) = P(s_t|s_{t-1}, s_{t-2}, \cdots, s_{t-j}) \tag{2.1}$$

When $j = 1$ we obtain a first order Markov process in which the the value of the process at times t depends only on the value at time $t-1$. In this case, the process is completely characterized by its first order state transition probabilities:

$$a_{ij} \doteq Pr(s_t = S_i | s_{t-1} = S_j) \tag{2.2}$$

2.1.2 Hidden Markov Models

Consider a dynamic system with a discrete finite state space. At time t, this system finds itself in one of N states and takes on a value generated by a state-dependent probabilistic distribution. An HMM is a model for a dynamic system in which a discrete Markov process is used to describe the dynamic properties or state evolution of the system, and state dependent probability distributions (discrete, continuous, or mixed) are used to model the observable outputs of the system. For this reason, HMMs are also known as doubly stochastic processes since there are two level of random processes in the underlying model: one which remains hidden describing the state occurrences, and another one at each state modeling the observable outputs of the system. The general structure of an HMM is shown in Fig. 2-1. The circles of the diagram indicate different states, and the arrows indicate probabilistic transitions between states. The squares indicate observable outputs from the HMM. Notice that there are two non-emitting states in this diagram. These states are reserved for the initial and final state of the model and allow the HMM to generate observations according to its own dynamics while ensuring that the initial and final states are always visited. The functionality of the non-emitting states becomes clear if we want to build composite models in which several single models are concatenated to model sequences which do not contain a single class, but rather several classes (as might be the case for a speech fragment containing several words, or a video sequence containing several facial expressions).

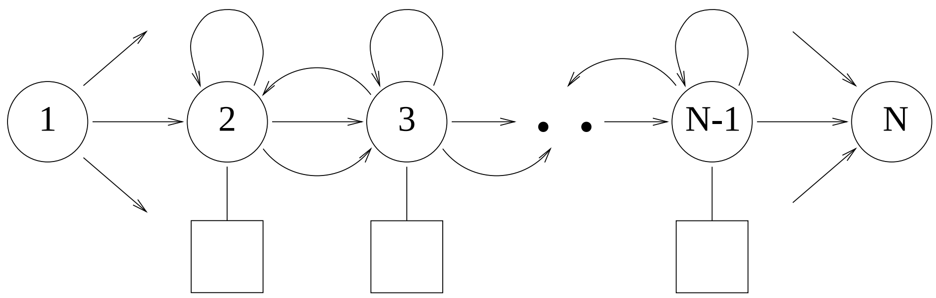

Figure 2-1: HMM with initial and final non-emitting states

For a first order HMM, the process of generating a dynamic system which follows the model consists of generating a state sequence according to the transition probabilities in 2.2 and then sampling from the output distribution associated with the state visited at time t. The problem of interest, however, usually consists of inferring the underlying model from a set of sequences which are assumed to have been generated by a common model. To this inference problem, we turn next.

2.2 Estimation

Let us define formally the parameters describing the above model. Consider that a set of M observation sequences $\{X^m\}_{m=1}^{M}$ is available, and let:

T_m be the length of the m^{th} observation sequence $\mathbf{X}^m = \mathbf{x}_1^m, \mathbf{x}_2^m, \cdots, \mathbf{x}_{T_m}^m$

N be the number of states in the model

$S = \{s\}$, $s_t = i$ denote state i at time t, and $i = 1, 2, \cdots, N$

$\pi = \{\pi_i | \pi_i = Pr(s_1 = i)\}$, initial state probabilities

$A = \{a_{ij}\}$ where a_{ij} is as defined in (2.2)

$B = \{f_i(\mathbf{x})\}$ where $f_i(\mathbf{x})$ is the probability density function associated with state i. In the most general case, we will model density functions with mixtures of K Gaussian-s such that

$$f_i(\mathbf{x}) = \sum_{k=1}^{K} c_{ik} \mathcal{N}(\mathbf{x}, \mu_{ik}, \Sigma_{ik}) \qquad \text{with} \quad \sum_{k} c_{ik} = 1 \quad i = 1, 2, \cdots, N \tag{2.3}$$

For a given HMM of order N, then the set of parameters $\theta = \{A, B, \pi\}$ define the model completely.

Given a set of observation sequences, there are three basic inference problems we need to address: the probability of *an* observation under the parameters of the model, how to modify the parameters of the model so as to maximize the probability of the data set, and lastly, what is the sequence of hidden states under the current model for the observation in question. We will address each one of these topics in the next sections:

2.2.1 The Forward-Backward Algorithm

The forward-backward algorithm addresses the first of these issues, namely how to compute the probability that an observation $\mathbf{x}_1, \cdots \mathbf{x}_T$ was produced by the model $\theta = \{A, B, \pi\}$. A direct method for computing the probability of an observation is through summation of all possible state sequences of length T:

$$Pr(\mathbf{X}|\theta) = \sum_{all\ S} Pr(\mathbf{X}, S|\theta) = \sum_{all\ S} Pr(\mathbf{X}^m|S, \theta) Pr(S|\theta) \tag{2.4}$$

for every fixed state sequence $S = s_1, s_2, \cdots, s_T$. Since for an HMM, we assume that the output observations are a function of only the current state, we obtain the following:

$$Pr(\mathbf{X}|S, \theta) = f_{s_1}(\mathbf{x}_1) f_{s_2}(\mathbf{x}_2) \cdots f_{s_T}(\mathbf{x}_T) \tag{2.5}$$

Likewise, for a first order HMM, the probability of the state sequence is

$$Pr(S|\theta) = \pi_{s_i} a_{s_1 s_2} \cdots a_{s_{T-1} s_T} = a_{s_0 s_1} a_{s_1 s_2} \cdots a_{s_{T-1} s_T} \tag{2.6}$$

Hence, (2.4) becomes

$$Pr(\mathbf{X}|\theta) = \sum_{all\ S} \prod_{t=1}^{T} a_{s_{t-1} s_t} f_{s_t}(\mathbf{x}_t) \tag{2.7}$$

Computing (2.7) requires the order of $O(N^T)$ operations, and as the number of states and, most importantly, the size of the observation grow in length, evaluating this expression becomes intractable. The forward-backward algorithm is an efficient implementation of (2.7) which requires considerably fewer operations (in the order of $O(N^2T)$) [14] . Let the forward variable $\alpha_t(i)$ be defined as

$$\alpha_t(i) \doteq Pr(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, s_t = i|\theta) \tag{2.8}$$

that is, the joint probability of the leading observation up to time t ending in state i.

Likewise, define the backward variable $\beta_t(i)$ as

$$\beta_t(i) \doteq Pr(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \cdots, \mathbf{x}_T | s_t = i, \theta) \tag{2.9}$$

that is, the probability of the trailing observation from time t onwards conditioned on the final state being i. The forward algorithm consists of the following recursion to find the probability of an observation [15], [14], [16]:
Initialization:

$$\begin{aligned} \alpha_1(1) &= 1 \\ \alpha_1(j) &= a_{1j} f_j(\mathbf{x}_1) \end{aligned} \tag{2.10}$$

Recursion: For $t = 2, 3, \cdots, T$, and for $j = 2, 3, \cdots, N-1$

$$\alpha_t(j) = [\sum_{i=2}^{N-1} \alpha_{t-1}(i) a_{ij}] f_i(\mathbf{x}_t) \tag{2.11}$$

Final Probability:

$$Pr(\mathbf{X}|\theta) = \alpha_T(N) = \sum_{i=2}^{N-1} \alpha_T(i) a_{iN} \tag{2.12}$$

Likewise, the probability of an observation can be obtained in terms of a recursion on the backward variable. This is the backward algorithm:
Initialization:

$$\beta_T(i) = a_{iN} \tag{2.13}$$

Recursion: For $t = T-1, T-2, \cdots, 1$, and for $j = 2, 3, \cdots, N-1$

$$\beta_t(j) = \sum_{i=2}^{N-1} a_{ji} f_i(\mathbf{x}_{t+1}) \beta_{t+1}(i) \tag{2.14}$$

Final Probability:

$$Pr(\mathbf{X}|\theta) = \beta_1(1) = \sum_{i=2}^{N-1} a_{1i} f_i(\mathbf{x}_1) \beta_1(i) \tag{2.15}$$

It is straightforward to show that the right hand sides of (2.12) and (2.15) are indeed equal to (2.7).

2.2.2 The Baum-Welch Re-estimation Algorithm for a Single Model

The most important problem in HMM modeling consists of learning the parameter set of a model. The search for the optimal parameters is made so as to maximize the likelihood of an observation set. There is no known maximum likelihood closed solution method for computing estimates of $\theta = \{A, B, \pi\}$. There exists, however an expectation-maximization gradient based iterative algorithm that searches for an optimal solution to this problem: the Baum-Welch re-estimation algorithm. The basis of the algorithm consists of evaluating the likelihood of the observations under the parameters of the model (using the forward-backward algorithm discussed above), and then proceed to adjust the parameter set in such a way that the likelihood is increased. This process can be repeated until it is estimated to converge at a local extremum. Since we are considering HMMs with output distributions as in (2.3), the set of parameters we need to learn is $\{a_{ij}\}, \{\pi_i\}, \{c_{ik}\}, \{\mu_{ik}\},$ and $\{\Sigma_{ik}\}$. Because we are considering HMMs with only one initial state, it follows that $\pi_i = \delta_{i-1}$ and this parameter needs no further estimation.

Let the following intermediate probabilities be defined as follows for the m^{th} observation sequence in the data set:

$$\begin{aligned}
\gamma_t^m(i,j) &\doteq Pr(s_t = i, s_{t+1} = j|\mathbf{X}^m, \theta) \\
&= \frac{Pr(\mathbf{X}^m, s_t = i, s_{t+1} = j|\theta)}{Pr(\mathbf{X}^m|\theta)} \\
&= \frac{\alpha_t^m(i) a_{ij} [\sum_{k=1}^{K} c_{jk} \mathcal{N}_{jk}(\mathbf{x}_{t+1}^m)] \beta_{t+1}^m(j)}{Pr(\mathbf{X}^m|\theta)} \quad \text{for} \quad 1 \le t \le T_m - 1
\end{aligned} \tag{2.16}$$

$$\begin{aligned}
\zeta_t^m(j,k) &\doteq Pr(s_t = j, k_t = k|\mathbf{X}^m, \theta) \\
&= \frac{Pr(\mathbf{X}^m, s_t = j, k_t = k|\theta)}{Pr(\mathbf{X}^m|\theta)} \\
&= \begin{cases} \frac{a_{1j} c_{jk} \mathcal{N}_{jk}(\mathbf{x}_t^m) \beta_t^m(j)}{Pr(\mathbf{X}^m|\theta)} & \text{for } t = 1 \\ \\ \frac{\sum_{i=2}^{N-1} \alpha_{t-1}^m(i) a_{ij} c_{jk} \mathcal{N}_{jk}(\mathbf{x}_t^m) \beta_t^m(j)}{Pr(\mathbf{X}^m|\theta)} & \text{for } 1 < t \le T_m \end{cases}
\end{aligned} \tag{2.17}$$

The denominators of (2.16) and (2.17) may be obtained from (2.12) or (2.15). Then the following re-estimates of the parameters are guaranteed to lead to an increase in the observation probability $Pr(\mathbf{X}|\theta)$ until a local maximum is reached [15], [14]:

$$\hat{a}_{ij} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T_m - 1} \gamma_t^m(i,j)}{\sum_{t=1}^{T_m - 1} \gamma_t^m(i)} \quad \text{for } 2 \le i, j \le N - 1 \tag{2.18}$$

$$\hat{a}_{1j} = \frac{1}{M} \sum_{m=1}^{M} \frac{\alpha_1^m(j) \beta_1^m(j)}{Pr(\mathbf{X}^m|\theta)} \quad \text{for } 2 \le j \le N - 1 \tag{2.19}$$

$$\hat{a}_{iN} = \frac{\sum_{m=1}^{M} \frac{1}{Pr(\mathbf{X}^m|\theta)} \alpha_T^m(i) \beta_T^m(i)}{\sum_{m=1}^{M} \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m} \alpha_t^m(i) \beta_t^m(i)} \quad \text{for } 2 \le i \le N - 1 \tag{2.20}$$

$$\hat{c}_{jk} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T} \zeta_t^m(j,k)}{\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_t^m(j)} \tag{2.21}$$

$$\hat{\mu}_{jk} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T} \zeta_t^m(j,k) \mathbf{x}_t^m}{\sum_{m=1}^{M} \sum_{t=1}^{T} \zeta_t^m(j,k)} \tag{2.22}$$

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \zeta_t^m(j,k) (\mathbf{x}_t^m - \hat{\mu}_{jk}) (\mathbf{x}_t^m - \hat{\mu}_{jk})^T}{\sum_{m=1}^{M} \sum_{t=1}^{T} \zeta_t^m(j,k)} \tag{2.23}$$

2.2.3 The Embedded Baum-Welch Re-estimation Algorithm

Given a set of training data to fit to a single HMM, (2.16)–(2.23) above may be used to estimate the parameters of such a model. What is often the case, though, is that we don't have access to separate data sets to train each model independently, but rather our data set contains data subsets corresponding to different categories we wish to model. More importantly, it is not always possible to robustly segment the data into different categories to then implement the single model training algorithm described above.

The embedded Baum-Welch algorithm [16] consists of a technique to train several models from a unique source of data by updating all models simultaneously. The algorithm works by linking several single models together to create a composite HMM which reflects the classification of different sections of the data set (i.e. different words in a continuous speech utterance, or several gestures linked temporally), and then proceeds to update the parameters of the composite HMM in a way similar to the algorithm described above. The non-emitting initial and final states that we considered in the HMM structure provide linking anchors when creating the composite HMM. The algorithm works by concatenating as many single models as there are classes to model in the date, evaluating the forward and backward probabilities of the composite model, and then re-estimating the parameters. The re-estimation algorithm is in essence similar to the one described for the case of single model re-estimation, with the exception that linking models together allows entry states to be occupied out of transitions from previous models whereas in the single model case the entry state is only visited at $t = 1$. Furthermore, transitions from entry states to exit states are allowed to bypass a model altogether.

To describe the changes necessary to the estimation equations for embedded training, let Q be the number of different models used in the composite HMM, and let N_q be the number of states in the q^{th} model. The forward-backward algorithm becomes the following [16]:
For $q = 1, 2, \cdots, Q$
<u>Initialization</u>:

$$\begin{aligned}
\alpha_1^{(q)}(1) &= \begin{cases} 1 & \text{if } l = 1 \\ \alpha_1^{(q-1)}(1) a_{1N_{q-1}}^{(q-1)} & \text{otherwise} \end{cases} \\
\alpha_1^{(q)}(j) &= a_{1j}^{(q)} f_j^{(q)}(\mathbf{x}_1) \\
\alpha_1^{(q)}(N_q) &= \sum_{i=2}^{N_q-1} \alpha_1^{(q)}(i) a_{iN_q}^{(l)} \qquad (2.24)
\end{aligned}$$

<u>Recursion</u>:

$$\begin{aligned}
\alpha_t^{(q)}(1) &= \begin{cases} 0 & \text{if } q = 1 \\ \alpha_{t-1}^{(q-1)}(N_q - 1) + \alpha_t^{(q-1)}(1) a_{1N_{q-1}}^{(q-1)} & \text{otherwise} \end{cases} \\
\alpha_t^{(q)}(j) &= \left[\alpha_t^{(q)}(1) a_{1j}^{(q)} + \sum_{i=2}^{N_q-1} \alpha_{t-1}^{(q)}(i) a_{ij}^{(q)} \right] f_j^{(q)}(\mathbf{x}_t) \\
\alpha_t^{(q)}(N_q) &= \sum_{i=2}^{N_q-1} \alpha_t^{(q)}(i) a_{iN_q}^{(q)} \qquad (2.25)
\end{aligned}$$

The backward algorithm for embedded training consists of the following recursion:
For $q = 1, 2, \cdots Q$

Initialization

$$
\begin{aligned}
\beta_T^{(q)}(N_q) &= \begin{cases} 1 & \text{if } q = Q \\ \beta_T^{(q+1)}(N_{q+1}) a_{1N_{q+1}}^{(q+1)} & \text{otherwise} \end{cases} \\
\beta_T^{(q)}(i) &= a_{iN_q}^{(q)} \beta_T^{(q)}(N_q) \\
\beta_T^{(q)}(1) &= \sum_{j=2}^{N_q-1} a_{ij}^{(q)} f_j^{(q)}(\mathbf{x}_T) \beta_T^{(q)}(j) \qquad (2.26)
\end{aligned}
$$

Recursion:

$$
\begin{aligned}
\beta_t^{(q)}(N_q) &= \begin{cases} 0 & \text{if } q = Q \\ \beta_{t+1}^{(q+1)}(1) + \beta_t^{(q+1)}(N_{q+1}) a_{1N_{q+1}}^{(q+1)} & \text{otherwise} \end{cases} \\
\beta_t^{(q)}(i) &= a_{1N_q}^{(q)} \beta_t^{(q)}(N_q) + \sum_{j=2}^{N_q-1} a_{ij}^{(q)} f_j^{(q)}(\mathbf{x}_{t+1}) \beta_{t+1}^{(q)}(j) \\
\beta_t^{(q)}(1) &= \sum_{j=2}^{N_q-1} a_{1j}^{(q)} f_j^{(q)}(\mathbf{x}_t) \beta_t^{(q)}(j) \qquad (2.27)
\end{aligned}
$$

The total probability (probability of the entire sequence) may be computed from either the forward or backward probabilities:
Final Probability

$$Pr(\mathbf{X}|\theta) = \alpha_T(N) = \beta_1(1) \qquad (2.28)$$

The re-estimation formulas for the state transition probabilities need to account for transitions between the single models of the composite HMM, and can be estimated as follows. Consider once again a set of M observation sequences $\{\mathbf{X}^m\}_{m=1}^M$ to each one of which we now want to fit Q single models. The rest of the parameters are defined as in section 1.2.2.

$$\hat{a}_{ij}^{(q)} = \frac{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m-1} \alpha_t^{(q)m}(i) a_{ij}^{(q)} f_j^{(q)}(\mathbf{x}_{t+1}^m) \beta_{t+1}^{(q)m}(j)}{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m} \alpha_t^{(q)m}(i) \beta_t^{(q)m}(i)}$$

for $2 \leq i, j \leq N-1$

$$\hat{a}_{1j}^{(q)} = \frac{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m-1} \alpha_t^{(q)m}(1) a_{1j}^{(q)} f_j^{(q)}(\mathbf{x}_t^m) \beta_t^{(q)m}(j)}{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m} \alpha_t^{(q)m}(1) \beta_t^{(q)m}(1) + \alpha_t^{(q)m}(1) a_{1N_q}^{(q)} \beta_t^{(q+1)m}(1)}$$

for $2 \leq j \leq N-1$

$$\hat{a}_{iN_q}^{(q)} = \frac{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m-1} \alpha_t^{(q)m}(i) a_{iN_q}^{(q)} \beta_t^{(q)m}(N_q)}{\sum_{m=1}^M \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m} \alpha_t^{(q)m}(i) \beta_t^{(q)m}(i)}$$

for $i \leq 2 \leq N-1$

$$\hat{a}_{1N_q}^{(q)} = \frac{\sum_{m=1}^{M} \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m-1} \alpha_t^{(q)m}(1) a_{1N_q}^{(q)} \beta_t^{(q+1)m}(1)}{\sum_{m=1}^{M} \frac{1}{Pr(\mathbf{X}^m|\theta)} \sum_{t=1}^{T_m} \alpha_t^{(q)m}(1)\beta_t^{(q)m}(1) + \alpha_t^{(q)m}(1) a_{1N_q}^{(q)} \beta_t^{(q+1)m}(1)} \tag{2.29}$$

2.2.4 Recognition. The Viterbi Algorithm

We next turn to the problem of using HMMs for classification of a sequence. In the case of single modeling, a classifier can be implemented, for instance, by use of Baye's rule to build a maximum *a posteriori* classifier which maximizes the posterior probability of a given model [17]:

$$\hat{\theta} = argmax_{\,\theta} \frac{Pr(\mathbf{X}|\theta)Pr(\theta)}{Pr(\mathbf{X})} \tag{2.30}$$

The main difficulty in evaluating (2.30) consists in evaluating the probability of an observation under a model $Pr(\mathbf{X}|\theta)$, which can be efficiently found through the forward-backward algorithm.

Given an HMM and an observation, an alternative way to implement a recognition system is to consider the most likely sequence of states that the observation followed through the model, and evaluate its likelihood. This consists of recovering the hidden part of the model from the observation, and then choosing the model with the most likely state sequence. Formally, we need to find the state sequence which maximizes $Pr(\mathbf{X}, S|\theta)$. The algorithm that allows us to recover the hidden state sequence is the Viterbi algorithm, and this method of classification is known as Viterbi decoding.

Since evaluating $Pr(\mathbf{X}|\theta)$ involves summing over all possible sequences of hidden states (see (2.4), the classifier in (2.30) will yield similar results to Viterbi decoding *if* we can approximate the sum over all state sequences by the most likely state sequence. Whereas we can choose either approach to classify a set of sequences assumed to be generated by a single model, it is the Viterbi approach that will allow us to classify sequences generated by different models at different regimes of the sequence, since the use of the total probability does not help detect changes in models. In other words, we need to solve the classification as well as the segmentation problem (establishing the boundaries between different models in a sequence).

The Viterbi algorithm is a recursive algorithm. Let the maximum joint probability of an observation and being in state i at time t be defined as follows:

$$\phi_t(i) \doteq Pr_{max}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, s_t = i|\theta) \tag{2.31}$$

Then the sequence of states and the maximum partial likelihood at each time step may be found through the following set of equations [14], [15]:
Initialization:

$$\begin{aligned} \phi_1(1) &= 1 \\ \phi_1(j) &= a_{1j} f_j(\mathbf{x}_1) \end{aligned} \tag{2.32}$$

Recursion: For $t = 2, 3, \cdots, T$, for $j = 2, \cdots, N-1$

$$\begin{aligned} \phi_t(j) &= max_{\,i}\{\phi_{t-1}(i)a_{ij}\} f_j(\mathbf{x}_t) \\ \psi_t(j) &= max_{\,i}\{\phi_{t-1}(i)a_{ij}\} \end{aligned} \tag{2.33}$$

Final Maximum Probability:

$$Pr_{max}(\mathbf{X}|\theta) = \phi_T(N) = max\,_i\{\phi_T(i)a_{iN}\} \tag{2.34}$$

State Recovery:

$$\begin{aligned} s_T^* &= N \\ s_{T-1}^* &= argmax\,_i \phi_{T-1}(i) \\ s_t^* &= \psi_{t+1}(s_{t+1}^*) \quad \text{for } t = T-1, \cdots, 2 \\ s_1^* &= 1 \end{aligned} \tag{2.35}$$

Let us now turn to the problem of recognizing a sequence that may contain data generated by one of several models. There are several existing approaches to decoding sequences containing connected models. One particular algorithm, known as the Token Passing algorithm, has been developed in the context of connected speech recognition with good reported recognition rates. The Token Passing algorithm [18] is basically a variant of the Viterbi algorithm above in which the initial and final states of the HMMs are used to record potential information about model boundaries. The standard Viterbi algorithm finds the optimal path through a "state-time" matrix by evaluating $\phi_t(i)$ for all times and states, and then backtracking to find the likelihood of the most likely state sequence. The token passing algorithm creates a linked record of information whenever a transition between models is detected, and then backtracks through the entire linked list to find the most likely series of model transitions, and discard the least likely inter-model transitions.

A token is a variable defined for each HMM which contains information regarding the partial likelihoods $\phi_t(i)$ as well as a link to a record of the composite HMM boundary information. The Token Passing algorithm implements a Viterbi decoder in parallel for each HMM under consideration updating the token variables accordingly. If an HMM exit occurs indicating a potential model boundary, a model boundary record is created to hold the token contents, the time instant, and the identity of the emitting HMM. The algorithm may be summarized as follows:

Initialization:

for $HMM = 1, 2, \cdots, L$

  Create a token variable containing the value of $\phi_1(i)$ and a link

  Initialize $\phi_1(i)$ as per (2.32)

Algorithm:

for $t = 1, 2, \cdots, T$

  for $HMM = 1, 2, \cdots, L$

    1. Evaluate $\phi_t(i)$

    2. Update the tokens

    If $j = argmax\,_i\,\phi_t(i) = N$

      1. Create a new boundary record containing

        * token contents

        * value of t

        * the label of the emitting HMM

      2. Update the token's link to point to this record

Back-tracing:

1. Trace back the model boundary records linked to the token with the highest probability
2. Extract the model boundaries and HMM identities

Figure 2-2 shows a diagram of how the algorithm works. Suppose that at time $t-2$, the last state of $HMM2$ is visited. This information is registered in a model boundary record, the token's link is updated (step 1) and the token is propagated into the HMM network. At time t, the same token which had last visited HMM2 now emerges from $HMM1$, this information is saved in another model boundary record, and the links updated as shown in steps 2 and 3.

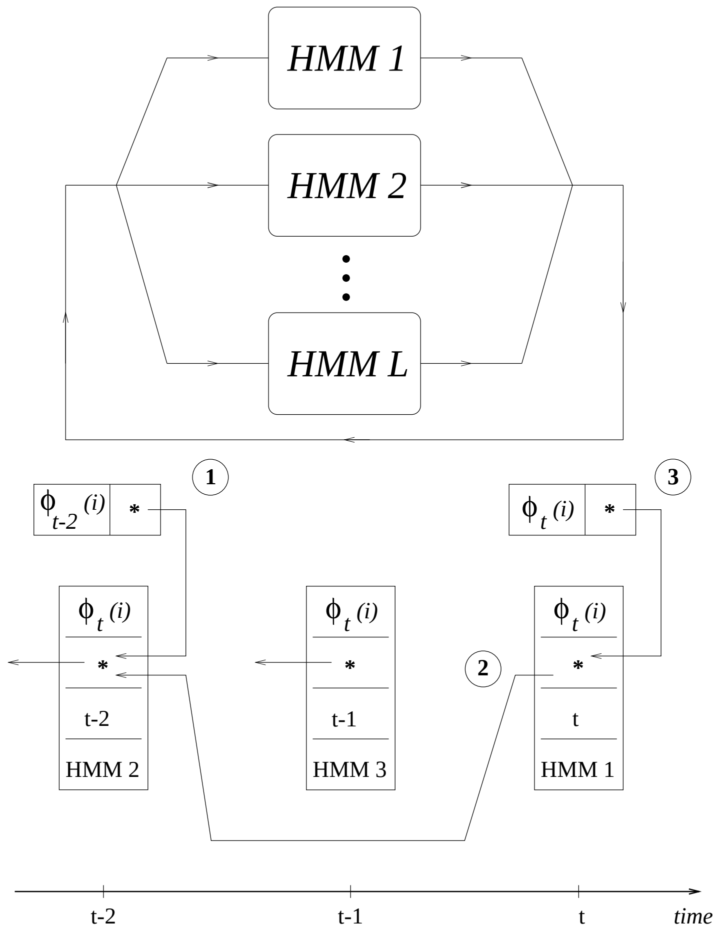

Figure 2-2: The Token Passing Algorithm

Chapter 3

Implementation

3.1 Establishing a Ground Truth

In a real application, a computer would have access to a set of biosignals which arrive in real time. To simulate this scenario, we implemented a data collection scheme in which the signals change over time as the subject is exposed to stimuli which we consider to elicit arousal. We wish to treat this problem as a classification problem and determine whether we can characterize and predict the instants of arousal from a set of observed physiological readings. Before proceeding to do this, a ground truth needs to be established in order to classify the observations. This is a non trivial problem which deserves careful consideration since the class categorizations we shall use to label the data have only been induced, not firmly established. In other words, there is an uncertainty associated with the class to which the data belongs. There is, for instance, a possibility that a stimulus failed to induce a high arousal response, and conversely, that a subject showed a high arousal response in the absence of the controlled stimulus due to another uncontrolled stimulus. In the classical recognition problem a set of data is used for learning the properties of the model under the different classes to recognize. The classification of this training data is usually fixed, and this knowledge is then used to derive the properties of the separate classes. We do not wish to abandon this framework entirely and will adopt a deterministic rule to label the training examples. However, establishing a proper labeling for the training data is one of the aspects of this problem which should be adaptive and subject to further discussion.

Our only degree of belief about what class the data belongs to is given by the onset of the pre-controlled stimuli during the course of the experiment. A rather intuitive approach to define the classes is to consider the response following a stimulus as representative of a "frustration" episode. How we establish the temporal segmentation following a stimulus deserves some attention. The time window we use to capture this response has to be wide enough to allow a *latency period*, as well as the true physiological response due to the stimulus. The latency period consists of the time lag which elapses between the onset of the stimulus and the start of the physiological change due to the stimulus. Some authors have established that for galvanic skin response this delay can be as much as 3 seconds [19]. The following diagram illustrates the principle used to label the data portion between any two stimuli: Figure 3-1 shows a portion of a mythical signal between two

stimuli represented by the bold vertical bars. Following the onset of the stimulus, we allow a dormant period of 1 second to pass before we start assigning the labels; then we window the following 10 seconds of data as representative samples of the class we want

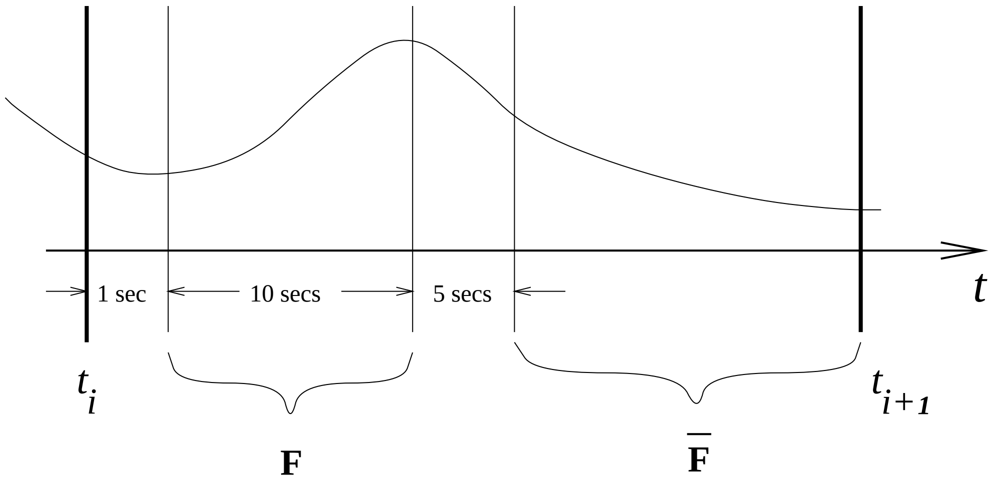

Figure 3-1: Labeling the data

to model as "*frustration*" (F). In order to transition out of this class, since the model boundaries are not known with precision, we allow another dormant period (of 5 seconds) without any classification, and then consider the rest of the signal up until the next stimulus to correspond to the class of "*non-frustration*" ($\overline{F}$). If the remaining set of samples is less than a minimum number of samples required to assign a label (3 seconds in these simulations), then a label is not assigned to this region. If the time windows used on two adjacent stimuli overlapped (the stimuli were spaced out by less than 10 seconds,) then the two resulting segments of data labeled as F would be merged together.

The chosen labels may be viewed as corresponding to positive and negative examples of the phenomenon we want to model. The reader should bear in mind, however, that this is a simplified mnemonic and modeling device and not an argument for what the true state of the physiology is since we can safely assume that human physiology exhibits much widely complex modes of behavior. The labeled regions roughly correspond to areas in which we have a higher degree of confidence about the class induced, whereas the unlabeled regions represent "don't-care" regions where our knowledge of the transition between affective states is too poor to include in the ground truth.

3.1.1 Stimulus Habituation and Anticipation

The procedure described above is the basis for establishing a ground truth which we used for the data analysis. There are several other variations which could result by simply adjusting some of the temporal parameters we used to define the labeling. How the variation of these parameters affects the overall classification performance is one of the research points we could extend beyond the work presented here. However, there is another aspect of the experiment which may condition the classification of the data, namely the user's expectation of a stimulus after a certain habituation period has elapsed. We decided to investigate whether there can be any effect on the performance of the models by redefining the ground truth in a way that accounts for a user's expectation of a stimulus. To do this, we adjoined to the set of actual stimuli a set of "virtual" stimuli where a user might have expected them. The idea would be, for instance, that if a user becomes aware that after T seconds from the first incident the mouse failed to work again, he might also anticipate a third failure approximately T seconds following the second failure (unless the actual third mouse failure happens first). Based on this, we used the following simple algorithm to augment the set of original stimuli. Let $\{t_i\}_{i=1}^{N}$ denote the set of time instants of the N original stimuli. Then:

$T = t_2 - t_1$
for $i = 3$ to N
  if $T < t_i - t_{i-1}$
    Insert a stimulus at $t = t_{i-1} + T$
  $T = t_i - t_{i-1}$

With the new set of augmented stimuli, we could then redefine the labels assigned to the data by following the procedure outlined in the previous section. The analysis of the data (model training and testing) can then be carried out independently according to each of the ground truths established in this section.

3.1.2 Other Alternatives

We have proposed an alternative ground truth to assign labels to the data. We mentioned earlier that other possibilities may be explored by adjusting some of the temporal parameters of the original labeling scheme. In particular, one alternative may be to model response times as a function of individual subjects since the goal of this work is to develop subject dependent systems, and physiological differences may be expected to arise across subjects.

Alternatively, if subjects habituate to the stimuli over time in a way that fails to elicit a response, it might be of interest to consider only a subset of the original set of stimuli and attach more uncertainty to the responses which occur after a habituation period has elapsed. This may be done by weighing the classification labels, or introducing a fuzzy classification to labels with high uncertainty.

Furthermore, we ought to incorporate knowledge about the underlying physiological behavior of these signals in modeling the ground truth. This will not only require us to import basic knowledge from the psychophysiology field, but also to take into account how the effect of the environment or the context in which the application takes place modifies the physiological behavior (i.e. detecting frustration in a highly stressful situation may differ from detecting it in a more relaxed context).

3.2 Feature Extraction

The biosignals collected during the experimental sessions consisted of galvanic skin response and blood volume pressure. A typical set of signals obtained from a subject is shown in Figure (3-2). The vertical bars overlayed on the plots indicate the onsets of the stimuli. From a set of raw data as the one shown in this figure, we need to obtain a set of significant feature signals that can bear some relevance to the recognition problem at hand, namely a set of features that might have correlates with internal affective states. This is one of the most important research problems that exist in this area: the mappings between affective states and physiological states is still an area which is being investigated at large in the psychophysiology community. In deciding on a feature set, we must account for classical measures of affective states (i.e. level of arousal as registered in a GSR signal, heart acceleration, etc), while bearing in mind that we can also allow the models we are

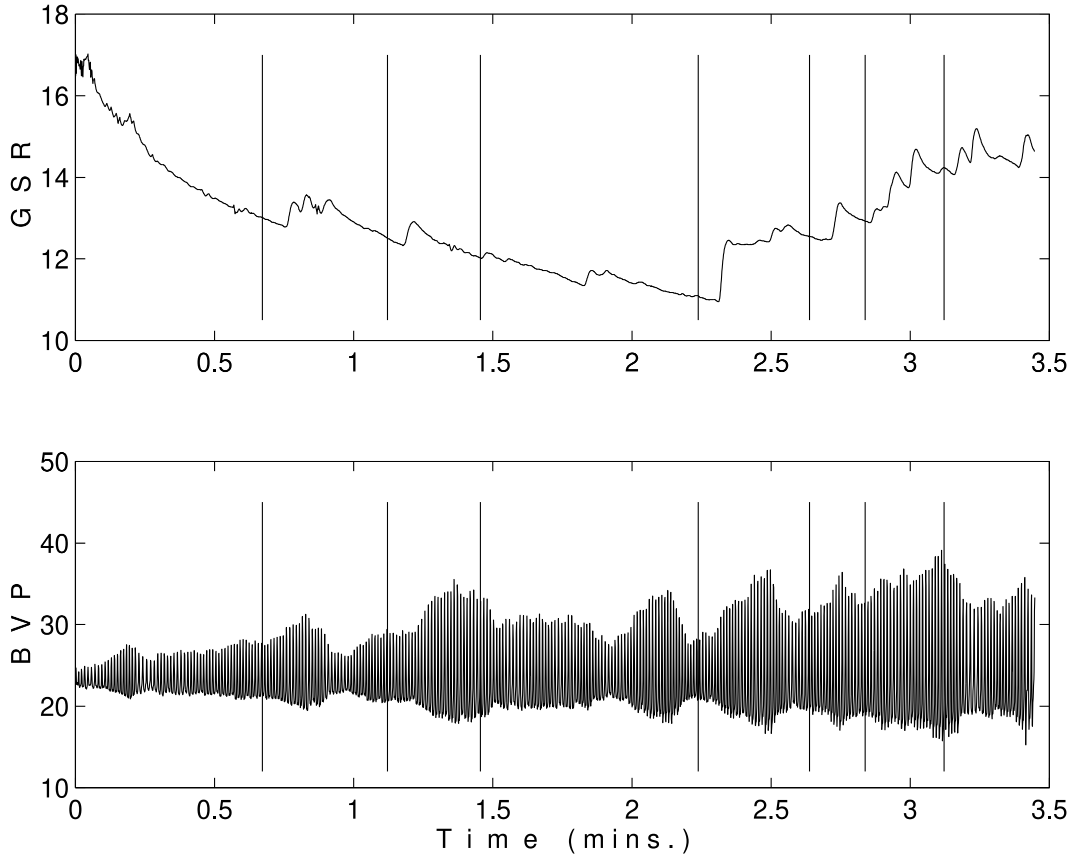

Figure 3-2: GSR and BVP signals

using to exploit more complex dynamic patterns that might not have received so much attention in other studies.

Let $g[n]$ and $b[n]$ represent the discrete time signals obtained by sampling the GSR and BVP signals. It is customary to measure changes in the GSR signal to predict levels of arousal. Motivated by this, we define the following signals:

$$g_\mu[n] \doteq g[n] - \frac{1}{N}\sum_{k=0}^{N-1} g[n-k] \tag{3.1}$$

$$g_v[n] \doteq \frac{1}{N-1}\sum_{k=0}^{N-1}\left(g[n-k] - \frac{1}{N}\sum_{l=0}^{N-1} g[n-l]\right)^2 \tag{3.2}$$

Equation (3.1) is just the GSR signal minus a time varying local sample mean obtained by windowing the GSR signal with an advancing N point rectangular window. This expression evaluates local fluctuations over time above a time varying mean value, which for a slow changing signal like the GSR, reveals local behavior of interest. Equation (3.2) is a time varying estimate of the local variance of the signal. It is obtained by windowing the GSR with an advancing N point rectangular window and evaluating the unbiased sample variance for every point.

Inspection of the BVP signal reveals that it exhibits a richer structure than the GSR signal. The BVP signal for instance has a richer harmonic content due to its periodic behavior over time. Its amplitude is also modulated in a way that we might exploit for feature extraction. In order to define suitable features on this signal, let us look at a portion of the BVP signal as shown in Figure 3-3. This figure shows several cycles of a BVP signal,

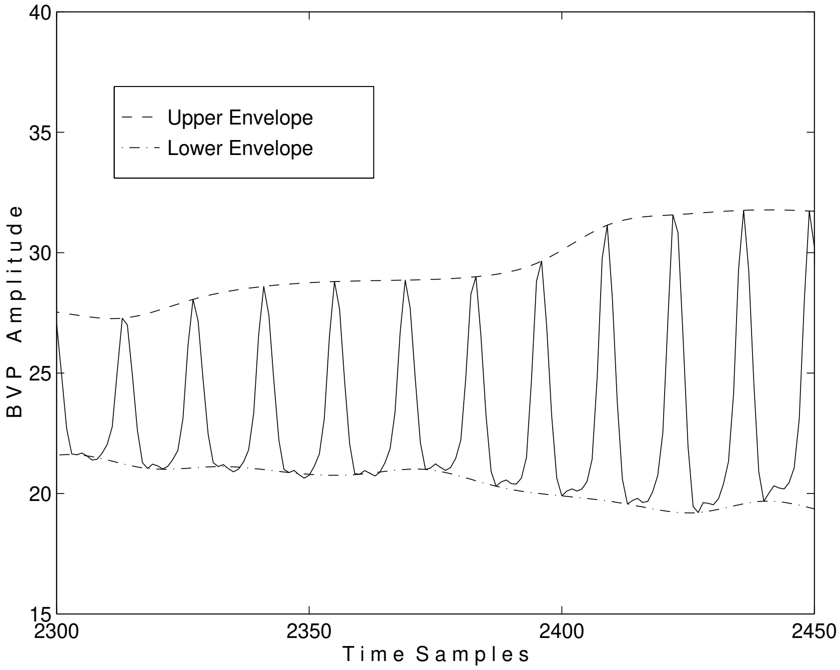

Figure 3-3: BVP signal

as well as a time varying upper and lower bound on the amplitude of the signal. Let $b_u[n]$ and $b_l[n]$ represent these two signals respectively. An efficient algorithm to find the upper envelope from the BVP is to find the peaks of the BVP signal and interpolate to obtain an upsampled waveform containing an equal number of points as the original BVP signal (in these plots, a signal containing the peak values was interpolated using cubic splines). Likewise, the lower envelope may be constructed by interpolating between the valleys of the BVP signal. Finding these peaks is straightforward by noting sign changes in the first difference above and below the local BVP mean.

Let us define the "pinch" of the BVP signal as the difference between these two envelopes:

$$b_p[n] \doteq b_u[n] - b_l[n] \tag{3.3}$$

Also, from finding the peaks of the BVP we can find the peak-to-peak intervals, and by taking this interval as one period of the harmonic oscillations, estimate local frequency as the reciprocal of the peak-to-peak intervals. Let $T_{p2p}[n]$ denote the number of samples between adjacent peaks. Again, we assume that $T_{p2p}[n]$ can be upsampled (i.e. cubic interpolation) to contain the same number of sample points across all signals. Experimentally, we found this simple method to agree with the results of the first harmonic obtained by short time Fourier analysis on the BVP signal, which is much more computationally intensive. By registering changes in the value of $T_{p2p}[n]$ (cycle duration), we can obtain an estimate of acceleration and deceleration of the harmonic cycles (this is a signal of interest since BVP is highly correlated with heart rate, and therefore so are changes in the BVP frequency).

Define then:

$$b_{\Delta T}[n] \doteq T_{p2p}[n] - T_{p2p}[n-1] \tag{3.4}$$

As noted earlier, the BVP frequency exhibits a rich behavior that can be characterized by changes over time as well as frequency. A way of studying this behavior is to observe its evolution in the time-frequency plane; one such approach was hinted at when we mentioned the short time Fourier transform. Another time-frequency approach to have received much attention lately, in particular in the study of non-stationary biosignals for feature extraction, is wavelet analysis [20]. The basics of wavelet decompositions are reviewed in Appendix B. Let us assume that an orthogonal wavelet decomposition of the BVP signal is implemented with J levels of resolution, and let

$$b(t) = \hat{b}^{(J-1)}(t) + \sum_{j=0}^{J-1} \sum_{k} d_{jk} w(2^j t - k) \tag{3.5}$$

where $\hat{b}^{(J-1)}(t)$ is the coarse approximation at level $J-1$, and $\{d_{jk}\}$ are the wavelet coefficients. The maximum level of resolution may be chosen empirically, so that the wavelet coefficients capture enough structure from the data. Wavelets are bases for continuous time functions. Since we don't have $b(t)$ but rather the set of samples $b[n]$, the implementation of a wavelet decomposition (i.e. via a filter bank scheme) will return an approximate set of wavelet coefficients $\{\hat{d}_{jk}\}$. The wavelet coefficients quantify the "detail" at a given level of resolution. The variability contained in this level of decomposition is reflected by the distribution of the wavelet coefficients; therefore, the expansion of a signal onto the basis at several scales followed by an analysis of the variance of the wavelet coefficients may be used to detect phenomena not easily observable in the original signal [20].

From the set of coefficients $\{\hat{d}_{jk}\}$ obtained through a filter bank decomposition, let us form a time series $d^{(j)}[n]$ by upsampling the coefficients at the j^{th} level to obtain a signal of equal length to the input signal. (Upsampling the wavelet coefficients doesn't convey much physical meaning. It is only done to be able to define a vector valued feature vector which will include the remaining features and exploit temporal structures between them). Similar to (3.2), we can obtain a local estimate of the variance of the wavelet coefficients by defining:

$$d_v^{(j)}[n] \doteq \frac{1}{M-1} \sum_{k=0}^{M-1} \left(d^{(j)}[n-k] - \frac{1}{M} \sum_{l=0}^{M-1} d^{(j)}[n-l] \right)^2 \tag{3.6}$$

and, using (3.1), (3.2), (3.3), (3.4), and (3.6), define the following 5-dimensional feature vector:

$$\mathbf{x}[n] \doteq \begin{bmatrix} g_\mu[n] \\ g_v[n] \\ b_p[n] \\ b_{\Delta T}[n] \\ d_v^{(j)}[n] \end{bmatrix} \tag{3.7}$$

In the implementation that follows, these are the values of the constants used in obtaining the features: $N = 200$ in (3.1) and (3.2) (windowing the GSR with a 10 second window), $M = 30$ in (3.6) (windowing with a 1.5 second window), $j = 3$ in (3.6) using Daubechies-4 orthogonal wavelets (see Appendix B).

Also, in order to avoid numerical errors (especially when estimating covariance matrices of very small values), the extracted features were scaled to exhibit a higher range of amplitudes. For these simulations we used the following scale factors: $[4,\ 10,\ 0.5,\ 500,\ 0.02]^T$. The values were chosen to keep the data within the range ± 2

3.3 Model Structure

In implementing a HMM as described in Chapter 2 for modeling and recognition, some of the parameters of the HMM were not subject to any sort of inference procedure in order to determine the optimal value from the data. For instance, the number of states N in a HMM must be chosen *a priori* before training begins, as is the number of Gaussian mixtures K in the state output distributions, and whether we choose full or diagonal matrices to model the covariance matrices of the distribution. Furthermore, even after N and K are chosen, we can impose restrictions on the transition probabilities so as to create HMMs that range from sparsely to fully connected.

Finding an optimal model topology is a difficult problem. Some of the existing approaches in the machine learning literature propose techniques that adaptively let the size of the model grow as needed; others start off with a size large enough to accommodate the data and resort to some sort of pruning techniques at the end (references would be nice). We will opt for a much simpler approach to selecting a structure in this study by evaluating the performance of a fixed subset of models. Since the goal of this research is to investigate the performance of *subject-dependent* systems, the search over the subset of interest in model space must be carried out for each subject independently.

One of the most widely studied HMM structures to model dynamic systems assumes that there exists a "causal" relation between the states visited. In this model, the states are an order set and are only allowed to be visited in "forward" or "causal" order. To allow this, the transition probabilities are constrained to satisfy $a_{ij} = 0$ for $i > j$ (the transition probability matrix A is upper triangular). Furthermore, the forward links may be restricted to be zero for all but a few states. These HMMs have been used at large in speech modeling, where the speech waveform is assumed to go through non-recurring states. Figure (3-4) shows the structure of one of these topologies. This topology is also known as "*left to right*" due to the structure of the transition probabilities. In this example,

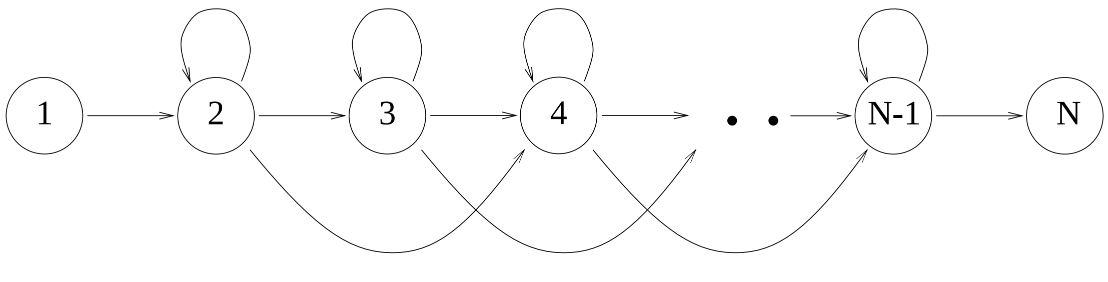

Figure 3-4: Left-to-Right HMM

transition probabilities are only nonzero for $\{a_{ij} \,|\, 2 \leq i < N-1; i \leq j \leq i+2 \leq N-1\}$ for all states except the initial and final states which only have one transition out of and into them respectively. To enforce this structure, the HMM's transition probability matrix must be initialized to have the appropriate entries being zero, since the Baum Welch re-estimation algorithm will not affect those entries.

Alternatively, we may consider HMMs in which, with the exception of the initial and final states which are singly linked, all transition probabilities are allowed to be non-zero, and states are allowed to be visited in a recurring fashion after a transition out of them has occurred. Such HMMs are known as ergodic, although transition probabilities which are initialized to non-zero values may be zero as a result of training.

To investigate the performance of different HMM types, we considered the following HMM classifications according to:

* number of states: $N \in \{4, 5, 6, 7\}$

* number of Gaussian components in output distribution: $K \in \{1, 2\}$

* type of covariance matrix Σ: diagonal, full

* transition probability type: causal, ergodic

The nonzero transition probability constraints for the causal HMMs were as follows:

$\{a_{ij} \,|\, 2 \leq i \leq N-1; i \leq j \leq i+2 \leq N-1\}$ for $N = 4, 5, 6$

$\{a_{ij} \,|\, 2 \leq i \leq N-1; i \leq j \leq i+3 \leq N-1\}$ for $N = 7$

in addition $a_{1j} = \delta_{j-2}$, and $a_{iN} = 0 \; \forall i \ni \{N-1, N\}$, for all HMM types

The resulting subset of models to train over consisted of 32 possible types ($4 \times 2 \times 2 \times 2$) for each subject. The results of the simulations are presented in the next chapter.

3.4 Software Implementation

The modeling of the data was carried out using the HMM modeling techniques described in Chapter 2 for the different HMM types proposed in the previous section. The software implementation of the learning algorithms was done using the Hidden Markov Model Toolkit (HTK) (version 2.0) developed at Cambridge University and Entropic Research Laboratories, Inc. The next steps were followed to train a model type from the data

1. Designate a subset of the subject's data for training the models, and reserve the remaining data for testing purposes.

2. Produce a label transcription (class identifier) following one of the ground truth schemes discussed in section 1.

3. For each HMM class, find initial estimates for the parameter set θ using the single model Baum Welch re-estimation algorithm (2.10) – (2.23) by extracting the data corresponding to the class under training from the compound observation sequence of the training set. This step serves as a *bootstrap* step to find good initial estimates of the parameters, which can be improved by performing embedded training next.

4. After the single model Baum-Welch algorithm has converged for the two classes we are modeling, apply the embedded training algorithm as described by (2.24) – (2.29) and update the parameters. A total of 6 entire iterations of the embedded Baum-Welch algorithm were used in these simulations.

5. Apply the token passing algorithm to the training *and* testing sets as described in section (1.2.4) and equations (2.32) – (2.34) to obtain a transcription of the recognized classes (label transcriptions).

6. Compare the recognition transcriptions to the original labels from step 2, and evaluate the results. If a label from step 2 contains a label of the same identity within its time boundaries, a successful classification is called; otherwise we designate a misclassification.

HTK has optimized libraries to carry out the computations required in steps 3 – 6 above. The implementation was carried out using this software package for the optimized performance required for running such a pool of subjects under the variety of conditions described above.

Chapter 4

Results and Discussion

In this section we present the results of the implementation described in the previous chapter. The data set consisted of 36 subjects who ran up to three experimental sessions during the data collection experiment. The subjects were given the choice of running a maximum of 3 sessions; however, many subjects decided to stop after the first session. Training models with these subjects' data presented a problem since not enough data was available for training and testing purposes. These subjects have been noted in the discussion of results that follows.

In order to assess the performance of the recognition system, we have evaluated the performance of the models for each of the labels we assigned to the data, as well as the overall recognition result. Even though our aim is to model *one* affective state we have described as "frustration", we trained the system by presenting it with positive (F) and negative ($\overline{F}$) examples of the class labels. A fair assessment of the performance of the system, therefore, requires that we examine how accurately the system assigns a positive or negative label to a portion of data. This is equivalent, if one views this problem as one of "frustration detection," to ensure that a good rate of detection is not obtained at the expense of a high rate of false alarms.

As discussed in Section 3.1, two ground truth labelings were considered in order to include a habituation effect. We approximated this effect by augmenting the original set of stimuli based on the periodic occurrences of the stimuli. In order to consider the validity of this labeling approach, we need to observe how the recognition results differ from those obtained with the standard ground truth.

The experimental sessions consisted of 5,4 and 7 stimuli respectively. For subjects that ran all 3 sessions, the first two sessions were used as training data and the last one as testing. For subjects that ran only two, the first session was used for training and the second for testing. For those subjects that only ran one session, we randomly selected either 2 stimuli for testing and 3 for training or vice versa. For the second set of ground truth, the number of stimuli was augmented for each subject's data. The total number of stimuli in this case could vary across subjects. The sessions used for training and testing, however, were kept the same as in the simulations run with the standard ground truth.

The following tables summarize the results for each of the subjects run. Table 4.1 shows the results obtained under the original label definitions; Table 4.2 shows those under the alternative labeling rules. Recognition rates over 50% are shown in bold.

Some of the sequential subject numbering breaks at a few points in the table. Subject numbers that are missing correspond to experimental sessions in which we ran into technical difficulties with the equipment (i.e. the sensors stopped registering after a certain point, one or more of the channels from the encoder was not reading in data, etc.). These subjects were left out of the final subject set.

One of the problems in modeling these data was to find a suitable model structure. A simple approach to investigate this was to establish a set of possible structures and train over this set. The last column in the result tables include the model topology with which the best overall training and testing results were obtained [1].

There was variability across subject-dependent model structure. However, the following remarks stand out. Causal HMMs were favored over ergodic topologies; this was particularly the case under the alternative labeling rules. The best results were obtained with 6, 5, and 7 (less frequently) state HMMs.

For this set of features, unimodal output distributions were more successful more frequently than bimodal distributions; full covariance matrices were as frequent as diagonal covariance matrices for the original data, whereas diagonal covariance matrices yielded the best results more often for the set of alternative labeling. With more data collected over time, we might be able to better assess the relative performance of one structure type over another

4.1 Discussion

Tables 4.1 and 4.2 reveal a variability of results for different subjects. In particular, the results for those subjects with insufficient data tend to suffer from the lack of training examples and the algorithm generalizes poorly to the testing set. For the tests run under the standard labeling rules, the means of the overall recognition rates are 71.19% for the training set and 55.75% for the testing set. If we exclude subjects with insufficient data from this evaluation, the mean figures obtained are 74.98% and 63.14% for the training and testing sets respectively.

For the simulations run under the alternative labeling rules, the means of the overall recognition rates are 66.47% and 52.44% for the training and testing sets (70.65% and 58.37% for the subset of subjects with sufficient data). Clearly, the overall recognition results degrade under the second set of labeling rules. This result suggests one of several possibilities, namely the validity of the original labeling scheme, the negligible impact of stimulus habituation during the experimental sessions, or the need for a more accurate way to model and incorporate the habituation effect into the labeling (recall that we use a rather simplified approach to modeling the occurrences of the stimuli). (Figures 4-1 and 4-2)

The performance for each individual label (F and $\overline{F}$) is shown in Figures 4-3 – 4-6. Here, histograms of the recognition results have been plotted under the two sets of labelings. Furthermore, we have separately considered the performance obtained for only those subjects with enough training data. This is shown in Figures 4-4 and 4-6.

By inspection of the histograms, we can readily observe the following.

- *Within* any given ground truth, the individual label performance improves when we redefine the set of subjects to exclude those with insufficient data (Figures 4-4 vs. 4-3, and 4-6 vs. 4-5).

[1]The HMM type C/E-Nn-D/F-Kk indicates whether the HMM had a causal or ergodic topology, the number of states n, whether the covariance matrix of the Gaussian output distributions was diagonal or full, and the number of Gaussian mixtures k used.

Subject	Training Set (%)			Testing Set (%)			HMM Type [‡]
	F	$\overline{F}$	Overall	F	$\overline{F}$	Overall	
Subject 1[†]	50.00	**66.67**	**60.00**	0.00	**100.00**	40.00	C-N6-F-K1
Subject 2	**66.67**	**80.00**	**72.73**	**75.00**	**80.00**	**77.78**	E-N7-F-K1
Subject 3	**66.67**	50.00	**57.69**	50.00	**60.00**	**55.56**	E-N5-F-K1
Subject 4	**66.67**	**55.56**	**61.11**	28.57	**83.33**	**53.85**	C-N5-F-K1
Subject 5[†]	50.00	**66.67**	**60.00**	0.00	**66.67**	33.33	C-N4-D-K2
Subject 6[†]	50.00	**100.00**	**75.00**	50.00	33.33	40.00	E-N5-D-K1
Subject 7	**75.00**	**80.00**	**77.78**	**60.00**	**60.00**	**60.00**	C-N4-F-K1
Subject 8	**83.33**	**58.33**	**70.83**	**75.00**	**60.00**	**66.67**	E-N7-F-K2
Subject 9[†]	**100.00**	50.00	**75.00**	33.33	50.00	40.00	C-N5-D-K1
Subject 10[†]	**100.00**	**100.00**	**100.00**	0.00	50.00	25.00	C-N7-D-K1
Subject 11	**75.00**	**100.00**	**88.89**	0.00	**66.67**	36.36	E-N5-D-K2
Subject 12[†]	50.00	**66.67**	**60.00**	0.00	**66.67**	33.33	C-N6-F-K1
Subject 13[†]	**66.67**	**100.00**	**83.33**	0.00	33.33	20.00	E-N5-F-K1
Subject 14	**58.33**	**69.23**	**64.00**	50.00	**60.00**	**55.56**	C-N6-D-K2
Subject 15	**80.00**	**83.33**	**81.82**	25.00	**100.00**	**66.67**	C-N6-F-K1
Subject 16	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	E-N6-F-K1
Subject 18[†]	50.00	50.00	50.00	**66.67**	50.00	**60.00**	C-N5-D-K1
Subject 19[†]	33.33	**66.67**	50.00	50.00	**66.67**	**60.00**	E-N5-D-K1
Subject 20	44.44	**72.73**	**60.00**	**57.14**	**62.50**	**60.00**	C-N7-D-K1
Subject 21	**66.67**	**60.00**	**63.16**	**57.14**	**62.50**	**60.00**	E-N7-D-K1
Subject 22	**80.00**	**100.00**	**90.00**	**75.00**	40.00	**55.56**	C-N5-D-K2
Subject 23	**80.00**	**75.00**	**77.78**	50.00	**60.00**	**55.56**	C-N6-D-K2
Subject 24[†]	33.33	**66.67**	50.00	0.00	**66.67**	40.00	C-N5-F-K2
Subject 25	**60.00**	**66.67**	**63.64**	50.00	**60.00**	**55.56**	E-N5-D-K1
Subject 26	**100.00**	**100.00**	**100.00**	**80.00**	**100.00**	**88.89**	E-N5-F-K1
Subject 27	**88.89**	**81.82**	**85.00**	42.86	**71.43**	**57.14**	C-N6-F-K2
Subject 29	**80.00**	**83.33**	**81.82**	50.00	40.00	44.44	C-N6-D-K1
Subject 30[†]	50.00	**66.67**	**60.00**	0.00	**100.00**	40.00	C-N6-F-K1
Subject 31	**85.71**	**87.50**	**86.67**	**66.67**	**100.00**	**71.43**	C-N6-D-K2
Subject 32	**66.67**	**63.64**	**65.00**	**71.43**	**66.67**	**69.23**	E-N5-D-K2
Subject 33	40.00	**66.67**	**54.55**	50.00	**100.00**	**77.78**	C-N4-D-K2
Subject 34	**55.56**	**70.00**	**63.16**	42.86	**60.00**	50.00	C-N5-F-K2
Subject 38[†]	33.33	50.00	40.00	**66.67**	50.00	**60.00**	C-N6-F-K1
Subject 39	**71.43**	**66.67**	**70.00**	**100.00**	50.00	**75.00**	C-N5-F-K1
Subject 41	**80.00**	**100.00**	**88.89**	50.00	**80.00**	**66.67**	C-N6-D-K1
Subject 44	**75.00**	**75.00**	**75.00**	0.00	**100.00**	**55.56**	C-N6-F-K2

[†] denotes subjects with insufficient data

[‡] see footnote on page 27

Table 4.1: Subject-dependent Recognition Results

Subject	Training Set (%)			Testing Set (%)			HMM Type [‡]
	F	$\overline{F}$	Overall	F	$\overline{F}$	Overall	
Subject 1[†]	50.00	50.00	50.00	**75.00**	33.33	**57.14**	E-N-7-D-K1
Subject 2	**76.47**	**83.33**	**79.31**	**60.00**	**66.67**	**63.64**	C-N6-D-K2
Subject 3	**64.71**	50.00	**57.14**	**66.67**	33.33	50.00	C-N6-D-K2
Subject 4	**61.54**	**90.00**	**73.91**	11.11	**66.67**	33.33	C-N7-D-K2
Subject 5[†]	**75.00**	**80.00**	**77.78**	0.00	50.00	25.00	C-N7-F-K2
Subject 6[†]	**66.67**	**100.00**	**83.33**	0.00	33.33	20.00	E-N6-D-K2
Subject 7	**100.00**	**100.00**	**100.00**	**62.50**	**57.14**	**60.00**	C-N4-D-K2
Subject 8	**58.82**	**71.43**	**64.52**	**80.00**	50.00	**63.64**	E-N7-F-K2
Subject 9[†]	**66.67**	50.00	**60.00**	50.00	0.00	40.00	C-N5-F-K1
Subject 10[†]	**75.00**	0.00	50.00	**66.67**	0.00	40.00	C-N5-D-K1
Subject 11	**60.00**	**83.33**	**72.73**	12.50	**88.89**	**52.94**	C-N4-F-K2
Subject 12[†]	50.00	**60.00**	**55.56**	25.00	**75.00**	50.00	C-N5-D-K2
Subject 13[†]	50.00	**60.00**	**55.56**	50.00	50.00	50.00	C-N6-D-K2
Subject 14	**68.75**	**66.67**	**67.74**	**60.00**	16.67	36.36	C-N7-D-K2
Subject 15	**75.00**	**55.56**	**64.71**	**60.00**	50.00	**54.45**	C-N4-F-K1
Subject 16	**66.67**	**75.00**	**70.00**	**100.00**	**100.00**	**100.00**	C-N7-F-K2
Subject 18[†]	**75.00**	50.00	**66.67**	33.33	50.00	40.00	C-N4-D-K1
Subject 19[†]	50.00	**60.00**	**55.56**	25.00	50.00	37.50	E-N6-D-K2
Subject 20	**53.85**	**53.33**	**53.57**	**75.00**	44.44	**58.82**	C-N4-F-K1
Subject 21	**76.92**	**61.54**	**69.23**	**55.56**	**55.56**	**55.56**	C-N7-D-K1
Subject 22	**75.00**	**100.00**	**86.67**	**60.00**	50.00	**54.55**	E-N5-F-K2
Subject 23	**75.00**	**75.00**	**75.00**	40.00	**83.33**	**63.64**	E-N5-F-K1
Subject 24[†]	50.00	**60.00**	**55.56**	25.00	50.00	37.50	C-N7-D-K2
Subject 25	**75.00**	**77.78**	**76.47**	**60.00**	**83.33**	**72.73**	C-N6-D-K2
Subject 26	**60.00**	**83.33**	**72.73**	**62.50**	**75.00**	**66.67**	C-N4-F-K2
Subject 27	**76.92**	**73.33**	**75.00**	**55.56**	28.57	43.75	C-N6-F-K2
Subject 29	**75.00**	44.44	**58.82**	**60.00**	50.00	**54.55**	E-N6-D-K1
Subject 30[†]	25.00	50.00	37.50	**66.67**	0.00	40.00	C-N7-D-K1
Subject 31	**60.00**	**66.67**	**62.50**	**71.43**	**100.00**	**75.00**	E-N7-D-K2
Subject 32	**61.54**	**66.67**	**64.29**	50.00	**66.67**	**57.14**	C-N4-F-K2
Subject 33	**62.50**	**77.78**	**70.59**	20.00	**100.00**	**63.64**	C-N6-D-K2
Subject 34	**61.54**	**76.92**	**69.23**	33.33	**66.67**	46.67	C-N7-D-K2
Subject 38[†]	50.00	50.00	50.00	**66.67**	0.00	50.00	E-N6-D-K2
Subject 39	**55.56**	50.00	**54.55**	**80.00**	25.00	**55.56**	C-N7-D-K1
Subject 41	**75.00**	**75.00**	**75.00**	**60.00**	**66.67**	**63.64**	C-N7-D-K1
Subject 44	**85.71**	**75.00**	**81.82**	**80.00**	33.33	**54.55**	C-N5-F-K2

[†] denotes subjects with insufficient data

[‡] see footnote on page 27

Table 4.2: Subject-dependent Recognition Results using Alternative Labeling Rules

- *Across* ground truth definitions, the performance for the two sets of subjects under distinction (all subjects, and those with sufficient data) tends to degrade under the alternative ground truth labeling (Figures 4-3 vs. 4-5 and 4-4 vs. 4-6).

The one exception to this last remark is the testing set of F labels. Recognition in this case improves under the alternative set of labeling rules. Since modifying the ground truth did not lead to an *overall* improvement in recognition for both training and testing sets, this improvement in performance on the testing set of F categories is not conclusive to assert the effectiveness of the alternative ground labeling.

It is also the case that the F label is consistently classified with a lower recognition rate than $\overline{F}$ for all the cases we have considered. This result only confirms the degree of uncertainty we have expressed about when to classify a portion of the data as representative of the class F. The inference about the existence of this category relies on the effectiveness of the inducing stimuli during the experimental sessions. One variation of an alternative ground truth which might be worth exploring further is one in which some of the stimuli are allowed to fail in their function to elicit a response. In other words, rather than augmenting the set of stimuli, we might be interested in studying the effects of reducing the original set of stimuli. This is one of many points of further research we propose in the next chapter.

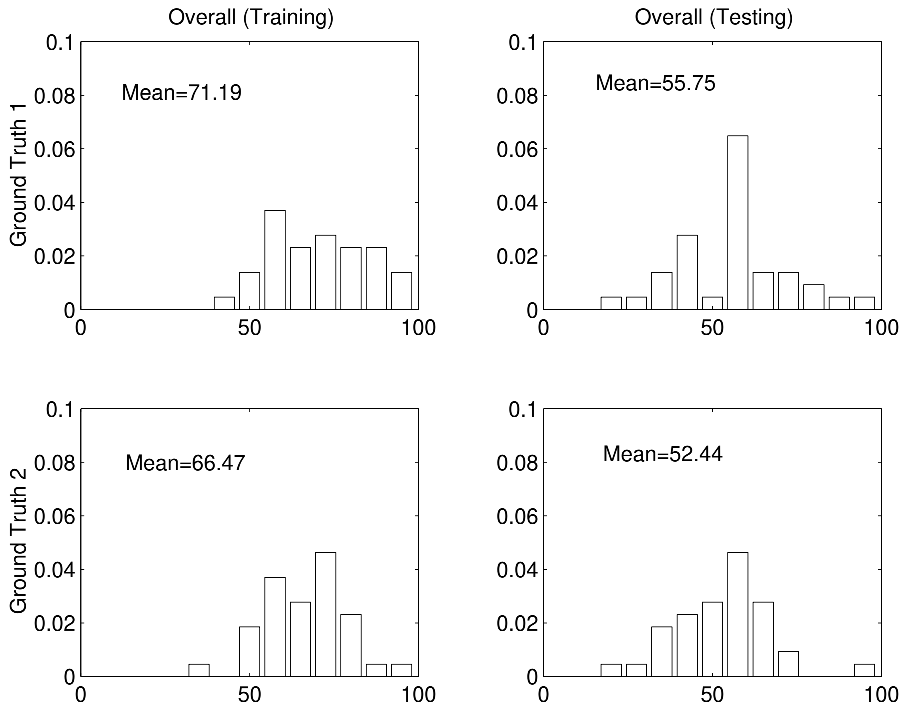

Figure 4-1: Histograms of Overall Recognition Results (all subjects considered)

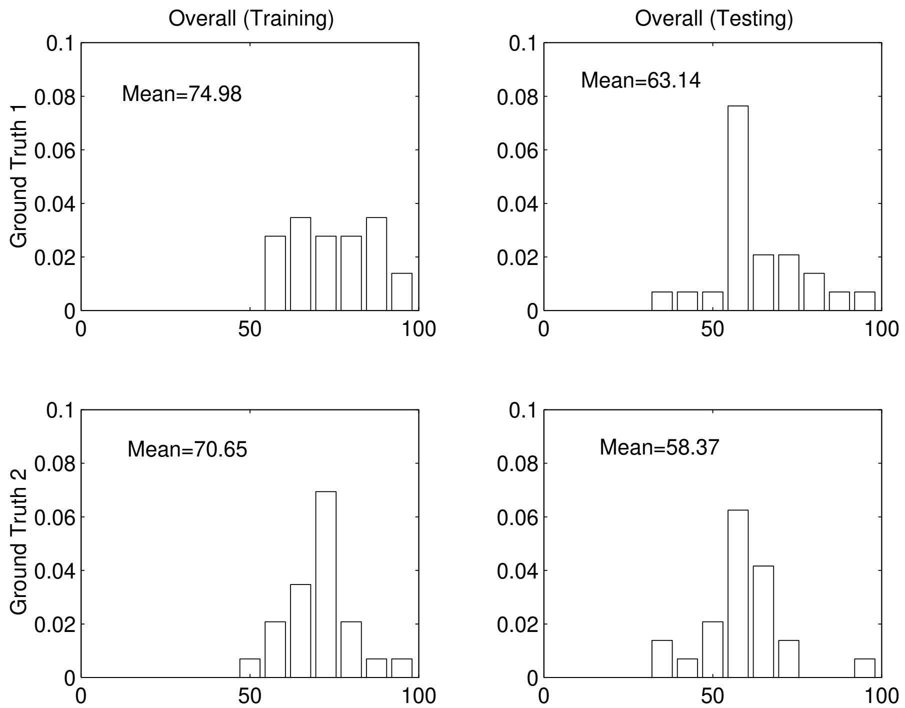

Figure 4-2: Histograms of Overall Recognition Results (subjects with insufficient data not considered)

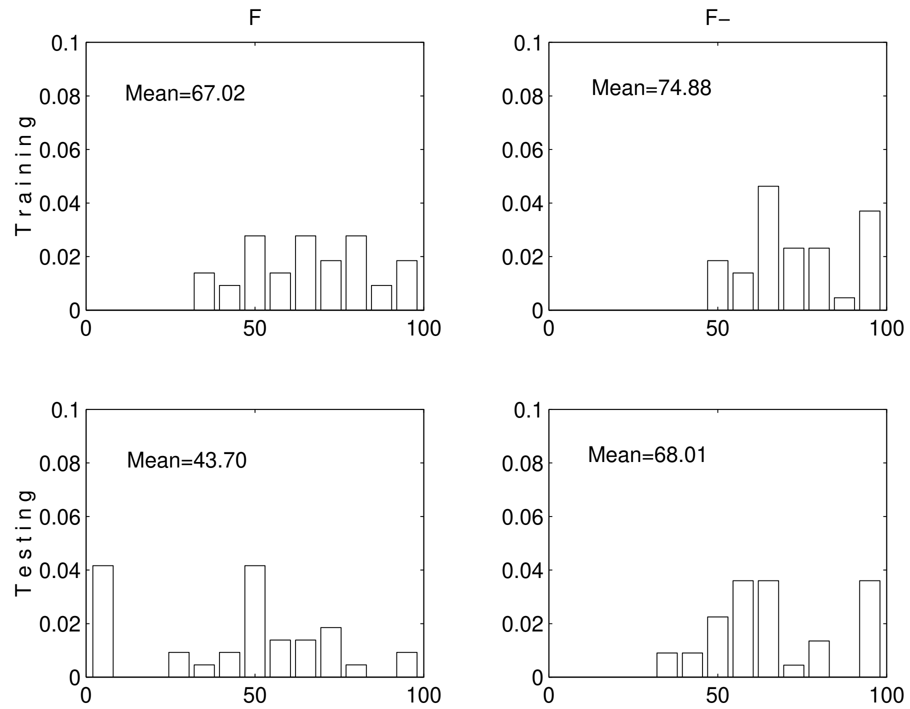

Figure 4-3: Histograms of Recognition Results (standard ground truth, all subjects considered)

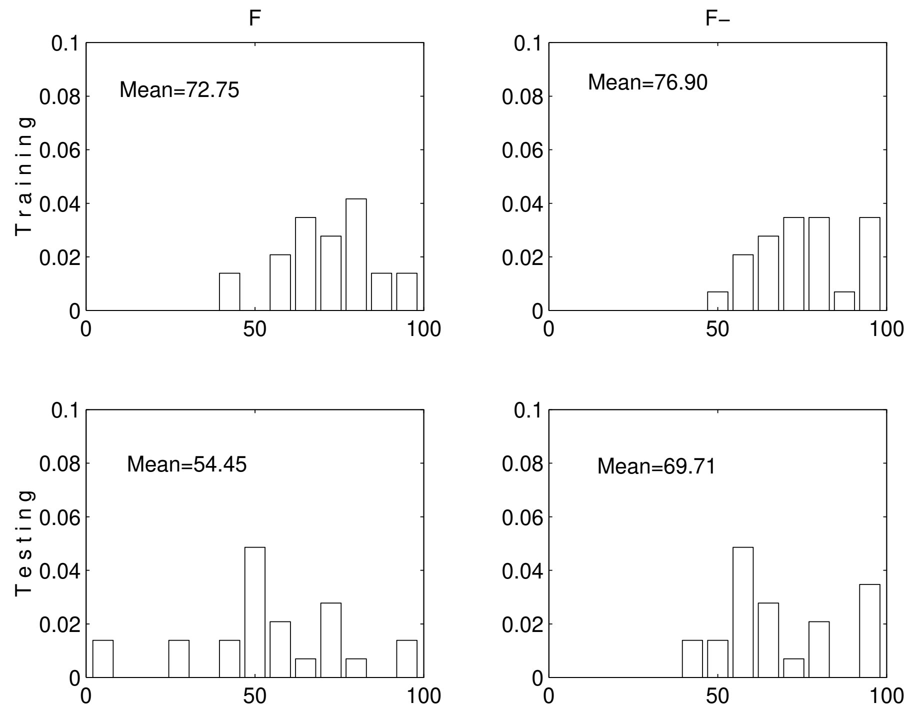

Figure 4-4: Histograms of Recognition Results (standard ground truth, subjects with insufficient data not considered)

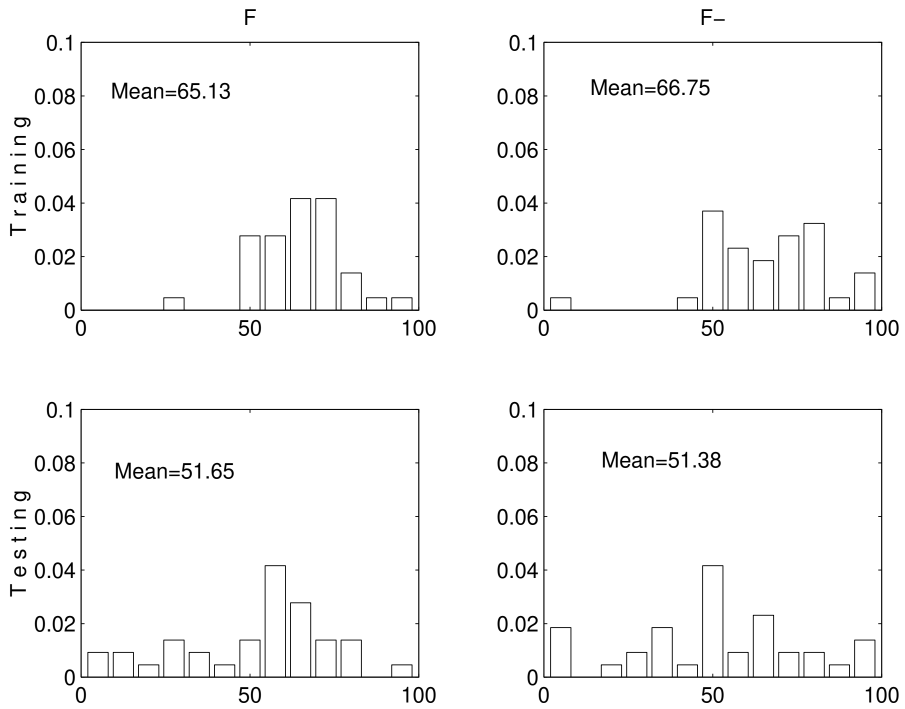

Figure 4-5: Histograms of Recognition Results (alternative ground truth, all subjects considered)

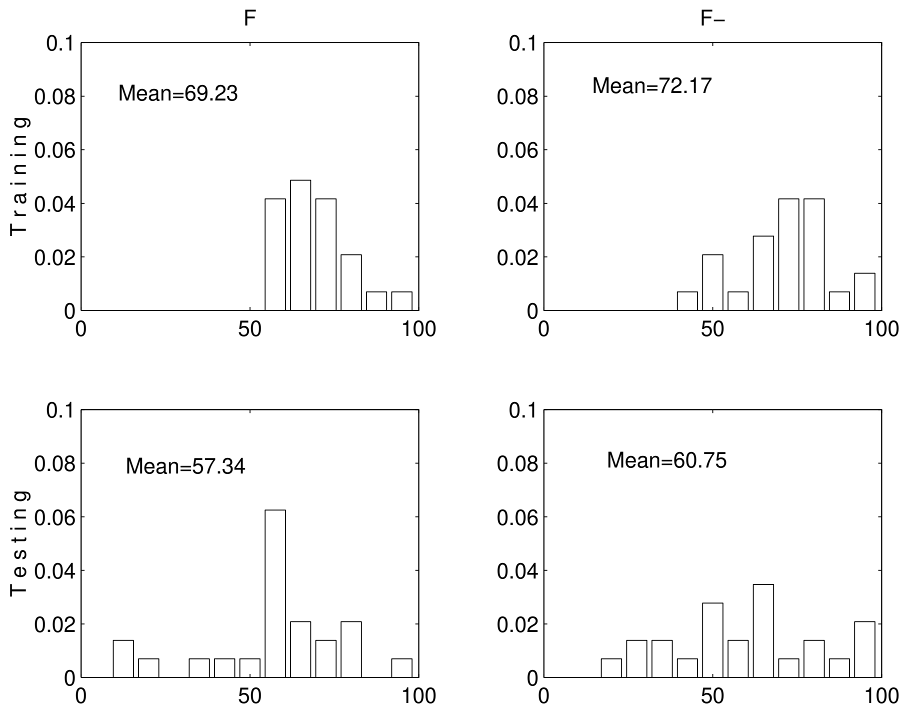

Figure 4-6: Histograms of Recognition Results (alternative ground truth, subject with insufficient data not considered)

Chapter 5

Conclusions

5.1 Summary

The work presented in this thesis constitutes one research effort in the area of recognition of human affect for affective computing applications. In particular, this work has approached the domain of human-machine interaction where there is arguably much room for improving the quality of human-computer interfaces. Motivated by the present inability of these interfaces to incorporate much of the affective nature of a human response into their system, we have explored the topic of recognition of human frustration as it arises when humans confront an interface which offers a faulty or inefficient design.

The analysis carried out in this thesis was based on data collected in a laboratory setting. A set of biosignals was collected from the subjects during the experimental sessions and subsequently used in a subject-dependent learning system to learn and predict patterns which corresponded to the presence or absence of the affective experience we wanted to model. A time series stochastic technique, Hidden Markov model, was used to implement the learning systems. Given the difficulties of establishing a reliable ground truth for classification, the need for evaluating the system performance under different criteria was discussed, and an alternative ground truth was proposed. The subjects' results were evaluated under two labeling criteria. The standard ground truth yielded better performance overall. When the entire data set was evaluated under this criterion, recognition rates greater than random were obtained for $\frac{2}{3}$ of the subjects. For many of the subjects, the amount of data collected was not enough to obtain good training and testing figures. When subjects with insufficient data were not considered in the evaluation rate, recognition rates greater than random were obtained for $\frac{7}{8}$ of the subjects.

5.2 Further Research

Affective computing is an area which still has many unsolved questions, and in which the challenges are numerous. This research project has helped gain insight into the many complexities that the design of affect recognition systems entails, and has suggested many possibilities as to what research directions we need to pursue next.

One of the most basic questions we can ask about this work is its extension to real scenarios that do not involve the laboratory constraints under which we have evaluated these data. There is a need to liberate this research from these constraints if we are to expect better generalizations. In this regard, non-invasive sensing systems that accompany a user during extended periods of time and collect data continuously might be an approach to take in data gathering. The role of the user is then fundamental to obtaining feedback regarding the state of the data in a way that can help us establish a possible ground truth.

The role of the user is also of great importance in building systems that learn over time. In this work, we have implemented a learning system which does not re-estimate its parameter after the entire learning phase is complete. This is partly due to the large amounts of data that such an approach would require. However, a learning system that obtains user feedback about its performance can use it to, for instance, re-evaluate not only its parameters, but also its ground truth about the data. We suggest that the evaluation of alternative ground truths can be performed in parallel and then user feedback about the performance can be used to promote or discard different labeling criteria.

Also, this research can greatly benefit from better understanding of the mappings between physiological responses and internal affective state. This suggests that we are interested not only in finding better features that carry affective content, but also in sensing more physiological signals. We have taken steps in this direction by also sensing the electromyographic response on the trapezius muscles for a few of the subjects from this data set. However, we still need to evaluate how this affects the performance of the system.

Extending the modeling techniques we have implemented here is certainly another open research topic. In particular, multi-modeling approaches have received attention for their ability to model different aspects of a complex problem separately and then combining them in an output decision. Developing other models which can perform efficiently on this data collection might be a next step to pursue in modeling affective signals.

Appendix A

Experimental Methodology

The subjects came to the lab to participate in an experiment that had been advertised on bulletin boards on campus. The subjects were informed that they were to participate in a visual perception experiment, and the real purpose of the experiment was not revealed to them until the debriefing period at the end of the experiment. The experiment consisted of a simple computer game in which there was a monetary reward motivation for superior performance. In order to create a very competitive environment, all subjects were made aware that this was a competitive task and that their reward was dependent on their performance with respect to the rest of the players. Performance was measured by a combination of the time required to complete the experiment and the number of errors incurred. The subjects were not told to what degree each of these factors influenced the final score and were encouraged to play as fast as they could. The actual task consisted of a simple interaction with a computer in which the user was presented with a series of slides containing multiple items of four different shapes and asked to indicate which shape contained the largest number of items. The only user interface allowed to the subjects was a mouse which they used to click on an icon corresponding to the desired answer. The simplicity of this "count and click" task is actually intended to keep the cognitive involvement of the subjects to a minimum since the role of cognitive load on the physiological response can be a confounding factor. The mouse was programmed to enter a delay mode at pre-specified intervals during the execution of the experiment. It is presupposed that, to the user, this created the impression that the mouse was simply failing to work at random points, thus interfering with his/her goal of finishing in a short time.

During the execution of the experiment, each subject's electrodermal response (GSR) and blood volume pressure (BVP response) were sensed, sampled at 20 Hz, and recorded [1]. To justify the presence of the invasive sensors during the experiment, the subjects were told that the purpose of the experiment was to study their physiological response to the visual stimuli appearing on the monitor. At the end, the subjects were asked to fill out standard questionnaires on demographic information and were subsequently debriefed. Some of the questions asked after the experiment addressed the degree to which the experiment had been successfully deceptive. If any subjects were to claim knowledge of the true purpose

[1]The sensing equipment is manufactured by *Thought Technology*. It consists of the sensors and and a ProComp encoder unit which samples the analog signals and then transfers the samples via fiber optic cable to an interfaced laptop computer. See `http://www-white.media.mit.edu/vismod/demos/affect/AC_research/sensing.html` for more details.

of the experiment prior to the debriefing period, they would be eliminated from the pool of subjects; however, this was not the case for any of the subjects who took part in the experiment.

Appendix B

Wavelet Decompositions

Wavelets are families of basis functions in $L_2(R)$. The classical definition of a wavelet also assumes that there exists a basic relation between the members of this family such that by the operations of scaling and dilation, all functions in the basis set may be obtained from a prototype called the "mother" wavelet. In the simplest case, the scaling operation is a dyadic scaling. If $\psi(t)$ denotes the prototype wavelet from a given family, then any function $f(x)$ in $L_2(R)$ may be written as:

$$f(x) = \sum_{j,k} d_{jk}\psi(2^j x - k) \tag{B.1}$$

Equation B.1 is the synthesis form of a wavelet expansion. If the wavelets are real and orthonormal, then obtaining the wavelets coefficients $\{d_{jk}\}$ from $f(x)$ (analysis) involves the same functions $\psi_{jk}(x)$ as in the synthesis equation.

The classical approach to wavelet construction involves finding the solution $\phi(x)$ to a two-scale equation:

$$\phi(x) = \sum_{k} h_k\phi(2x - k) \tag{B.2}$$

Then, for the case of orthogonal wavelets with finite support ($k \in [0, 1, \cdots, N-1]$ in B.2), the prototype wavelet may be obtained directly from $\phi(x)$ by [21]

$$\psi(x) = \sum_{k} (-1)^k h_{N-k}\phi(2x - k) \tag{B.3}$$

This method of construction is based on a multiresolution analysis approach. If the coefficients $\{h_k\}$ and $\{(-1)^k h_{1-k}\}$ which satisfy (B.2) and (B.3) are known, then a fast implementation of a wavelet decomposition can be achieved with an iterative filter bank scheme (filtering and downsampling). Equation (B.1) writes $f(x)$ as a combination of wavelet functions at all scales j. A filter bank implementation, however, can only execute a finite number of iterations and return a decomposition at J levels of resolution ($j = 0, 1, \cdots, J-1$). Equation (B.1) may then be written as:

$$f(x) = \hat{f}(x) + \sum_{j=0}^{J-1}\sum_{k} d_{jk}\psi(2^j x - k) \tag{B.4}$$

where $\hat{f}(x)$ is now a coarse approximation to the original signal $f(x)$, or the "remainder" of the signal which was not decomposed at levels of resolution beyond $J-1$

In the filter bank implementation, a set of low and high pass filters is obtained from the coefficients $\{h_k\}$ and $\{(-1)^k h_{1-k}\}$. Then, at step i of the iteration, the coefficients at the output of the low pass analysis branch of the filter bank (low pass filtered and downsampled by 2) from the $i-1$ step are reused as inputs to the analysis filter bank. At the i^{th} step, the coefficients from the high pass branch are saved as the wavelet decomposition coefficients. To properly initialize this algorithm (at $i=0$), the coefficients going into the filter bank should be obtained by evaluating the inner products of $f(x)$ with $\phi_{0,k}(x)$. Often times, $f(x)$ is not known directly but through a set of samples $f[n]$. In this case evaluating the inner products may be not be so straightforward, and a suboptimal approach is to use the samples $f[n]$ as the actual inputs to the filter bank to initialize the algorithm [21].

To analyze the BVP data collected in this experiment, we used Daubechies-4 orthogonal wavelets. The "mother" wavelet at scale $j=0$ is shown in Figure B-1. In choosing a suitable wavelet family to analyze the data, we tried to select a wavelet that captures some of the characteristics of the data, such as smoothness or oscillatory behavior.

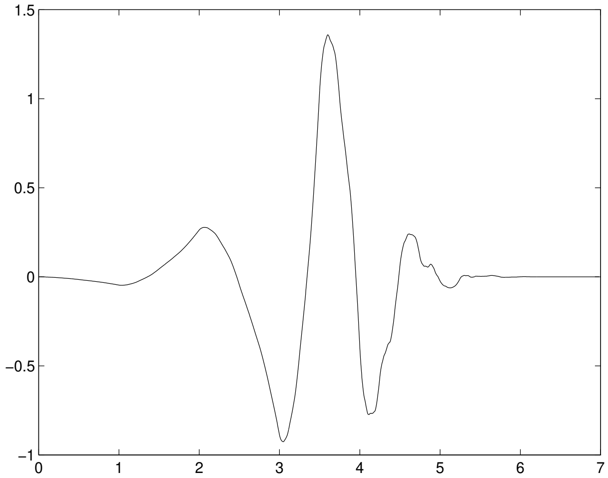

Figure B-1: Daubechies-4 Wavelet

The filter bank implementation uses the following 8-tap FIR filter obtained from the 8 coefficients satisfying B.2 for the Daubechies-4 wavelet [22]

$$h_k = \{-0.0106\ \ 0.0329\ \ 0.0308\ \ -0.1870\ \ -0.0280\ \ 0.6309\ \ 0.7148\ \ 0.2304\} \tag{B.5}$$

The maximum level of resolution used in the decomposition was $J=4$. This was empirically adjusted by selecting a resolution level which captured most of the detail in the signal.

Bibliography

[1] R. W. Picard. *Affective Computing.* The MIT Press, Cambridge, Massachusetts, 1997.

[2] A. R. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain.* Gosset/Putnam Press, New York, NY, 1994.

[3] J. LeDoux. *The Emotional Brain.* Simon & Schuster, 1996.

[4] C. vanOyen Witvliet, and S. R. Vrana. Psychophysiological responses as indices of affective dimensions. *Psychophysiology*, 32(5):436–443, 1995.

[5] S. R. Vrana. The psychophysiology of disgust: Differentiating negative emotional contexts with facial EMG. *Psychophysiology*, 30(3):279–286, 1993.

[6] S. W. Porges. Cardiac vagal tone: A physiological index of stress. *Neuroscience and Biobehavioral Reviews*, 19(2):225–233, 1995.

[7] J. T. Cacioppo, and L. G. Tassinary. Inferring psychological significance from physiological signals. *American Psychologist*, 45(1):16–28, January 1990.

[8] A. T. Pope, E. H. Bogart, and D. S. Bartolome. Biocybernetic system evaluates indices of operator engagement in automated task. *Biological Psychology*, 40(1):187–195, 1995.

[9] S. Makeig, T-P. Jung, and T. J. Sejnowski. Using feedforward neural networks to monitor alertness from changes in EEG correlation and coherence. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editor, *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference*, pages 931–937, Cambridge, Massachusetts, 1996. M.I.T. Press.

[10] A. Sloman, and M. Croucher. Why robots will have emotions. In *Seventh Int. Conf. on AI*, pages 197–202, Aug. 1981.

[11] I. A. Essa. *Analysis, Interpretation and Synthesis of Facial Expressions.* PhD thesis, M.I.T. Media Lab, Feb. 1995.

[12] I. R. Murray, and J. L. Arnott. Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *Journal of the Acoustic Society of America*, 93:1097–1108, Feb. 1993.

[13] H. Schlosberg. Three dimensions of emotions. *Psychological Review*, pages 81–88, March 1954.

[14] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition.* Information Technology Series. Edinburgh University Press, Edinburgh, 1990.

[15] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 1986.

[16] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *HTK - Hidden Markov Model Toolkit.* Entropic Research Laboratory, Inc.

[17] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley & Sons, New York, 1973.

[18] S. J. Young, N. H. Rusell and J. H. S. Thornton. Token passing: a simple conceptual model for connected speech recognition systems. Technical report, Cambridge University Engineering Department, 1989.

[19] M. Helander. Applicability of driver's electrodermal response to the design of the traffic environment. *Journal of Applied Psychology*, 63(4):481–488, 1978.

[20] M. Akay. Wavelet applications in medicine. *IEEE Spectrum*, pages 50–56, May 1997.

[21] Gilbert Strang, and Truong Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge, 1996.

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