State-based Classification of Finger Gestures from Electromyographic Signals

Peter Ju PJU@MIT.EDU
Research Laboratory of Electronics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA

Leslie Pack Kaelbling LPK@AI.MIT.EDU
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139 USA

Yoram Singer SINGER@CS.HUJI.AC.IL
School of Computer Science & Engineering, The Hebrew University, Jerusalem 91904, Israel

Abstract

Electromyographic signals may provide an important new class of user interface for consumer electronics. In order to make such interfaces effective, it will be crucial to map EMG signals to user gestures in real time. The mapping from signals to gestures will vary from user to user, so it must be acquired adaptively. In this paper, we describe and compare three methods for static classification of EMG signals. We then go on to explore methods for adapting the classifiers over time and for sequential analysis of the gesture stream by combining the static classification algorithm with a hidden Markov model. We conclude with an evaluation of the combined model on an unsegmented stream of gestures.

1. Introduction

In the past, as computers became more powerful, new interfaces were needed to take advantage of that power. Recently, digital devices have tended towards portability as well as increased in speed. The demand for smaller cellular phones and personal digital assistants is proof of this trend. Consequently, new interfaces now have to accommodate portability as well as power.

A new input device has been invented by researchers at AT&T Research that enables a user to control portable devices with simple small finger gestures. Electromyographic (EMG) signals are recorded using a wrist-worn input device. When a user performs a finger gesture, such as touching index finger to thumb, the EMG signal is sent to a nearby device using a small-field wireless broadcaster. The device classifies the signal, mapping it to the gesture performed, and then carries out the action associated with the gesture.

In order to make such interfaces effective, it will be crucial to map EMG signals to user gestures in real time. The mapping from signals to gestures will vary from user to user, so it must be acquired adaptively. In addition, from time to time for a single user, the position of the electrodes will vary, requiring the classifier to be adapted online.

In this paper, we describe and compare three methods for static classification of EMG signals, then go on to explore methods for adapting the classifiers over time and for sequential analysis of the gesture stream.

2. Background and Previous Work

When a muscle is voluntarily activated, the sum of the action potentials from motor neurons and muscle fibers associated with the contracting muscle forms a voltage or EMG signal. This EMG signal can propagate through fat, tissue, and skin. As a result, a surface electrode placed on the skin can detect these signals. The characteristics of surface EMG signals have been studied extensively. Typically, an EMG signal is modeled as a slowly varying stochastic process. The majority of the signal energy occurs under 500 Hz with energy peaking at around 50-150 Hz. The voltage can range from 0 to 10 mV peak-to-peak. For short periods of time (around 0.1 seconds), EMG signals are stationary.

Electromyographic signals have been studied in the past, predominantly addressing the design of prosthetics for amputees. Graupe and Cline Graupe and Cline (1975) were the first to classify EMG signals with reasonable success. They used an autoregressive (AR) model of the EMG signal and were able to classify a single channel of data with an 85% success rate. Doerschuck et al. Doerschuck et al. (1983) improved upon Graupe and Cline's traditional AR approach for EMG signal classification by using the cross-correlation information between the EMG signals of multiple electrodes. Instead of using the traditional AR coefficients for EMG signal classification, Kang et al. Kang et al. (1995) explored the use of cepstral coefficients as a feature set. In their experiments, the AR method demonstrated a 78.7% classification rate, and the cepstral coefficient method demonstrated an 83.7% classification rate when classifying ten distinct gestures with four channels of recorded data. Englehart et al. Englehart et al. (1999) attempted to classify EMG signals using various time-frequency based representations with dimensionality reduction. In their experiments, Englehart et al. achieved a maximum classification rate of 93.7% using the coiflet wavelet packet family as a feature set, principal components analysis for feature selection, and linear discriminative analysis for classification. They were classifying four gestures using two channels of data.

The classification results from these experiments look very promising. However, their experimental set-up differs considerably from ours. Since these studies were attempting to create controls for prosthetic limbs, they focused on large movements using large muscle groups. Our goal is to try to classify small finger gestures controlled by a complicated set of small muscle groups using a highly unobtrusive sensor placed in a location that is convenient for the user (the wrist), but highly suboptimal for signal recording.

In pilot experiments, we implemented the methods described by Kang et al. Kang et al. (1995) and Englehart et al. Englehart et al. (1999), and found that their performance on our problem was quite poor. This motivated the development of additional classification methods, as well as the incorporation of dynamic training and sequential analysis to improve performance.

3. Static Classification

The first problem we addressed was simple supervised classification of two-channel signals into one of four classes. The signals were represented as sequences of real-valued samples of the EMG signal generated during a particular gesture. We implemented and tested three classification methods: a simple nearest-neighbor method on unprocessed data, a variation of Doerschuck's AR technique, and a new method based on iterative scaling.

3.1 Nearest Neighbor

As a baseline system, we implemented a simple nearest-neighbor classifier, using cross-correlation between signals as the distance metric. In the static classification setting we are given a set of training signals $\bar{x}_k = (x_k[1], x_k[2], \ldots, x_k[n])$, each of length N, and associated classes c_k. For a test signal $\bar{u}$, we compute an inverse distance between the test and training signals as

$$D(\bar{x}_k, \bar{u}) = \sum_{n=1}^{N} x_k[n]\, u[n] \; .$$

The class assigned to signal $\bar{u}$ is the class, c_k, of the signal $\bar{x}_k$ for which $D(\bar{x}_k, \bar{u})$ is maximized. In order to remove errors due to imprecise segmentation of the signals, we actually calculated the cross-correlation between each $\bar{x}_k$ and many shifts of $\bar{u}$, and then assigned to signal $\bar{u}$ the class of $\bar{x}_k$ for which $D(\bar{x}_k, \bar{u})$ is maximized. The comparison of shifts of the same signal can be done efficiently using the fast Fourier transform in $N \log(N)$ time per test signal (Porat, 1997).

3.2 Autoregressive Models

The second approach we tested was the standard method for classifying EMG signals, which is to build autoregressive (AR) models from the training data of each class, then classify new signals according to how far they deviate from the AR models of the classes. The AR parameter estimation technique tries to fit data to an all-pole filter excited by a spectrally flat source. It is useful when dealing with systems that are comprised of a simple source driving a complex time varying filter such as speech. EMG signals can be thought of as the result of such a source-filter system. The source would be impulse trains originating from the motor cortex, and the filter would be the system of neurons and muscle through which the signal is propagated. A pth-order AR model is described by the equation

$$x[n] = \sum_{i=1}^{p} a_i x[n-i] + e_\sigma[n] \; .$$

where again $\bar{x}$ is the signal, $\bar{e}$ is zero-mean white noise with variance σ^2, and $a_1 \ldots a_p$ are the AR coefficients.

The first step is to build AR models for each class from the training signals associated with that class. The AR coefficients are estimated from the training signals using a least squared error estimation method to minimize the difference between the actual signals and those that would have been generated by the model,

$$\sum_k \sum_{n=1}^{N} (x_k^j[n] - \sum_{i=1}^{p} a_i^j x_k^j[n-i])^2 \quad ,$$

where $\bar{x}_k^j$ ranges over training signals for a class j. Once this is done, a variance for each class, σ_j^2, can be directly estimated.

After AR models have been estimated for every class, to classify a signal, $\bar{u}$, we compute an estimated signal, $\hat{u}^j$, for each class j. This signal contains the predicted values, at each time step, that signal $\bar{u}$ would have in expectation, if it were generated by the AR model for class j,

$$\hat{u}^j[n] = \sum_{i=1}^{p} a_i^j u[n-i] \ .$$

Given the white noise assumptions, the negative log likelihood of $\bar{u}$ with respect to class j is,

$$L(\bar{u}|j) = N \log \sqrt{2\pi\sigma_j^2} + \sum_{n=1}^{N} \frac{(\hat{u}^j[n] - u[n])^2}{2\sigma_j^2} \ .$$

Finally, we assign signal $\bar{u}$ the class j for which $L(\bar{u}|j)$ is minimized.

The above scheme describes the one-channel method that Graupe et al. used to classify EMG signals. In our setting we have two channels of data, and as a result we employed a multi-channel extension of Graupe's method developed by Doerschuck et al. Doerschuk et al. (1983). In a general M-channel AR model, the signal at time n is an M dimensional vector $X[n]$ described by the vector equation

$$\sum_{i=1}^{p} \overbrace{\begin{bmatrix} a_{1,1}^i & \cdots & a_{1,M}^i \\ \vdots & \vdots & \vdots \\ a_{M,1}^i & \cdots & a_{M,M}^i \end{bmatrix}}^{A^i} \underbrace{\begin{bmatrix} x_1[n-i] \\ \vdots \\ x_M[n-i] \end{bmatrix}}_{X[n-i]} + \underbrace{\begin{bmatrix} e_1[n] \\ \vdots \\ e_M[n] \end{bmatrix}}_{E[n]} ,$$

where $A^1 \ldots A^p$ are the AR matrices and $E[n]$ is a vector of white noise processes. Specifying the rest of the model and classifying with it is analogous to the single channel case. However, the single variance parameter is replaced by a covariance matrix, and a multivariate Gaussian distribution is used to calculate the likelihoods.

3.3 Iterative Scaling

Auto-regressive models provide a simple and efficient approach for the problem of classification of *stationary* signals. However, the conditional probability model induced by auto-regressive modeling is deficient. This is mainly due to the fact that we are using a *generative* approach. Hence, the AR models for the different gestures might concentrate on stochastic characteristics that are *common* to all gestures. The class probability estimates are computed (using Bayes' rule) from the individual AR models, resulting in probability-"smeared" estimates.

We now describe a *discriminative* approach that directly estimates the class-conditional probabilities $P(y|x)$. Our approach is based on a parameter estimation technique called iterative scaling (IS) (Csiszar & Tusnady, 1984; Della Pietra et al., 1997). Our classification scheme also borrows ideas from wavelet basis selection (Coifman & Wickerhauser, 1992; Mallat & Zhang, 1993; Chen et al., 1998).

To build an estimate for the class-conditional probabilities we use a conditional exponential model of the form

$$P_{\bar{\alpha}}(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{l} \alpha_i f_i(x,y)\right) \ . \tag{1}$$

For concreteness, let $y \in \{1,\ldots,4\}$ be the class index. The functions $f_i(x,y)$ are called features and can take real values. We discuss below methods for choosing features. α_i is a real-valued weight associated with f_i. We denote by $\bar{\alpha}$ the vector of weights $(\alpha_1,\ldots,\alpha_l)$. $Z(x)$ is a normalization constant ensuring that $P(y|x)$ sums to one, $Z(x) = \sum_{y=1}^{4} \exp\left(\sum_{i=1}^{l} \alpha_i f_i(x,y)\right)$. Given a training set of signal-gesture pairs $\{x_k,y_k\}_{k=1}^{m}$, the log-likelihood of the training set for a particular value of $\bar{\alpha}$ is

$$\begin{aligned} L(\alpha) &= \sum_{k=1}^{m} \log\left(P_{\bar{\alpha}}(y_k|x_k)\right) \\ &= \sum_{k=1}^{m}\sum_{i=1}^{l} \alpha_i f_i(x_k,y_k) - \log(Z(x_k)) \ . \end{aligned}$$

Note that the log-likelihood is non-linear due to the term $\log(Z(\cdot))$. Therefore, it is not possible to have an analytical form for the vector of weights $\bar{\alpha}$ which maximizes the empirical log-likelihood. Iterative scaling performs additive updates to the weights, $\alpha_i \rightarrow \alpha_i + \delta_i$, guaranteeing that the log-liklihood is increased on each iteration. In our application we used a recent variant of iterative scaling, called improved iterative scaling (Della Pietra et al., 1997), for which there is a closed form solution for the update vector $\bar{\delta}$. Let C be a constant, defined below; then the update is

$$\delta_i = \frac{1}{C}\log(Q_i) \ , \tag{2}$$

where

$$Q_i = \frac{\sum_k f_i(x_k,y_k)}{\sum_k \sum_{y=1}^{4} P_{\bar{\alpha}}(y|x_k) f_i(x_k,y)} \ . \tag{3}$$

(Note that the value of $\bar{\delta}$ depends on the current probability estimates $P_{\bar{\alpha}}(y|x)$.) It is possible to show that this iterative process converges to the *global* maximum of the log-likelihood.

It remains to describe our choice of features for finger gesture classification. Normally, iterative scaling uses features that are binary functions from an instance-class pair (x,y) onto a 1 or 0. However, it can be shown (details omitted due to the lack of space) that iterative scaling will also converge using real-valued features. To achieve a closed form update given by Eqs. (2) and (3), the features need to satisfy the following constraints:

$$\forall x,y: \quad f_i(x,y) \geq 0 \quad \text{and} \quad \sum_{i=1}^{l} f_i(x,y) = C \ . \tag{4}$$

In other words, all the feature values must be positive, and for every instance-class pair, the features must sum to the same value. In practice, the first constraint can be satisfied by subtracting the minimum feature value (over the training set) from all the features. The second constraint can be satisfied by adding a new *virtual* feature that forces all the features to sum up to a single constant:

$$\tilde{f}(x,y) = C - \sum_{i=1}^{l} f_i(x,y),$$

where $C = \max_{x,y} \sum_{i=1}^{l} f_i(x,y)$ is set in order to satisfy the conditions given by Eq. (4). It is straightforward to verify that any weighted combination of both f_i's and $\tilde{f}$ is still linear in f_i's. Therefore, the resulting model induces a conditional class probability as given by Eq. (1).

For our features, we experimented with various time-frequency decompositions including long and short time Fourier transform, and various wavelet families, of the raw EMG signal. After transforming each channel of the EMG signal, we take the transform's coefficients and feed them as real-valued features to iterative scaling. In the case that we have four classes, since each coefficient is a scalar, we need to expand it to a 4-component vector in order to to use it as a feature f_i in the form given by Eq. (1). For brevity and concreteness let us drop the feature index i and assume that c is a transform coefficient of an EMG signal for a gesture from the second class. Then, we define a feature f based on c as follows,

$$c \quad \mapsto \quad f \stackrel{\text{def}}{=} (0, c, 0, 0) \ .$$

In other words, a transform coefficient c generated from a signal from class j defines a feature f that "abstains" on all the classes but the class j for which its prediction is of "strength" c. This construction is reminiscent of the specialists model used in on-line prediction (Blum, 1997; Freund et al., 1997) and weak hypotheses that can abstain in the context of boosting (Schapire & Singer, 1999). This particular form forces each feature to extend predictions only for the class from which it was derived.

In the experiments reported below, we used the first $N/2$ coefficients of the N-point discrete Fourier transform of each of the two channels of data, giving a total of N features. Since the signals are real, the DFT is symmetric. Therefore, we only need half the coefficients. In order to classify a new EMG signal, we first project the signal onto the feature space, expand the projection as described above, and then weight the features according to the weights assigned by IS. We classify the signal as belonging to the class which has the highest likelihood value.

4. Adaptation

Due to slight shifts in the position of the EMG recorder and changes in the environment and physiology of the user, a classifier trained on one day may not perform well on data collected from a different day. One possible mode for gathering additional training data is to ask the user to go through a short sequence of calibration gestures each time the device is placed on the wrist. These calibration gestures will provide an additional set of classified training examples.

One obvious strategy would be to save the original training data, add this new data, and re-run the batch training algorithm. But we assume that the device is not capable of storing all previous data, and we require that the adaptation in reaction to the calibration data be very efficient.

In our implementation, we used a stochastic gradient descent rule to adapt the weights of the AR model [1] to minimize the mean squared error. The loss function for the AR weights for a class j is therefore,

$$L_2(\bar{a}) = \sum_{t=1}^{T} (x[t] - \sum_{i=1}^{p} a_i^j x[t-i])^2 \ \ ,$$

where x is an EMG signal sequence belonging class j. Taking the partial derivative of the loss function with respect to individual weights, we can derive the following update rule for weight i in class j,

$$a_i^j \leftarrow a_i^j - \eta \sum_{i=1}^{p} \left(a_i^j x[t-i] - x[t] \right) x[t-i] \ \ ,$$

where η is a learning rate. The larger the learning rate, the more emphasis is placed on the new data. This update is due to Widrow and Hoff Widrow and Hoff (1960). Theoretically, in order for the Widrow-Hoff adaptation algorithm to converge to batch performance, $\sum_{t=1}^{\infty} \eta^2$ must converge to a finite value. ¿From empirical studies, it has been shown that a learning rate $\eta = 1/(t^{1+\delta})$ works well in practice. In our experiments, we used a learning rate of $1/(t^{1.2})$.

In addition to updating the AR coefficients, we also need to update the estimates for the variances. Assume we have encountered t signals for class j so far, the update for σ_j^2 is,

$$\sigma_j^2 \leftarrow \frac{t}{t+1}\sigma_j^2 + \frac{\xi}{t+1} \ \ ,$$

where

$$\xi = \frac{\sum_{n=1}^{N} (\sum_{i=1}^{p} a_i^j x_{t+1}[n-i] - x_{t+1}[n])^2}{N} \ \ .$$

[1] Since the AR technique is a generative classifier, it is easier to incorporate it into the adaptation and HMM sequence decoding techniques. However, in future research we will look at incorporate the IS classifier into these techniques as well.

5. Sequence Decoding

In a real application of the EMG device, the classifier will not be given isolated segments of the signal to classify. Rather, it will be given a sequence of voltage samples recorded from a sequence of gestures, for which it will have to produce a classification at every sample. With the AR method, sample classifications are produced by finding the estimated value at a given sample by weighting the previous p samples with the AR coefficients for each class:

$$\hat{x}^k[n] = \sum_{i=1}^{p} a_i^k x[n-i] \; .$$

Then, the likelihood of an observation $x[n]$ given that it was generated by the model for class k is

$$P(x[n]|k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(\frac{(\hat{x}^k[n] - x[n])^2}{2\sigma_k^2}\right) \; .$$

We can then classify a sample as belonging to the class with the highest probability. However, this approach does not incorporate timing information, which can help classification. When a person taps a finger, the smallest duration for that gesture is approximately 50 to 100 milliseconds. If we are sampling at 1 kHz, the smallest duration for a gesture is 75 to 100 samples, and therefore the probability for a change in gesture is at most 1 out of 75. The naive approach does not take into account the history of the signal, making its classification results very noisy, jumping from state to state in rapid succession.

Doerschuck et al. Doerschuk et al. (1983) provide an alternative method for calculating class probabilities at each time step. Let $p_k[n]$ denote the probability that gesture k is taking place at time step n. Then, using Bayes' rule, we can calculate $p_k[n]$ using the following recursive equation,

$$p_k[n] = \frac{P(x[n]|k)\, p_k[n-1]}{\sum_{k=1}^{4} P_k(x[n]|k)\, p_k[n-1]} \; ,$$

and defining $p_k[0] = 1/4$. From these class probability estimates, we report the class that has the maximum probability at each time step. Information about the history of the signal is only propagated through the conditional class probability estimates $p_k[n-1]$.

The problem with this method is that it assumes that the whole sequence up to and including the sample you want to classify is one gesture. It assumes that if you are in a state then it is very probable that you will stay in that state. However, even though the assumption is a close approximation to reality, it is not correct and causes problems. In order to switch states, you have to build up enough evidence to prove that the entire sequence up to the sample being classified is the new gesture, and not the previous gesture. This results in a significant delay in the estimated onset time of a new gesture.

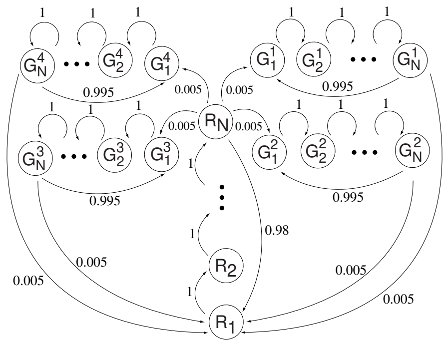

Figure 1. The HMM model used for sequence decoding. State R_i is rest timing state i; state G_i^j is gesture j timing state i.

In order to incorporate timing and state transition information into our sample classification system, we combined a very simple hidden Markov model (HMM) with the AR classification model. The HMM model used in the experiment is shown in Fig. 1. It consists of five major groups of states: a rest group, and one group for each gesture. Each group contains 50 timing states. Since no gesture lasts for less than 50 samples, the timing states are used to ensure that the best state sequence is relatively smooth, and only allow the sequence to exit a state every 50 samples. The observation probabilities for this HMM are calculated using the same method as in the naive sequence decoding method. The Viterbi algorithm is then used to determine the most probable state sequence given the data.

Currently, the model parameters are manually set because we do not have any sequences of gestures collected in a task setting. In future research we plan to use the Baum-Welch algorithm (Baum & Eagon, 1967) to tune the model parameters based on actual sequences generated in particular tasks or by particular users.

6. Data Collection

We used Delsys Inc.'s Bagnoli-2 EMG System for data collection. It consists of two active differential surface electrodes, one main amplifier unit, and one reference electrode cable. The maximum output voltage range is 5 volts, and the channel frequency response is 20 Hz to 500 Hz. The electrodes were positioned around the right wrist of the subjects. One electrode was positioned on the palm side of the arm, approximately an inch below the wrist joint, on the pinky-half of the hand. The other electrode was placed on center of the non-palm side of the arm, approximately an inch below the wrist joint. We sampled the signals at 1024 samples/sec for 2 seconds per gesture. The signals were then segmented into examples of length 512 samples.

The EMG signal is affected greatly by the subject's fat and tissue content. Therefore, we needed retrieve data from a wide range of body types. The study had a total of 21 subjects: four female and seventeen male. Their body weights ranged from 140 to 200 lbs, and their ages ranged from 22 to 50. During a data collection session, a subject performed four gestures 15 times each. The four gestures were: a press of the index finger to the the thumb, a press of the middle finger to the thumb, a press of the ring finger to the thumb, and a press of the pinky finger to the thumb. Three subjects participated in one data collection session per day over the course of eight days in order to determine classification rate across sessions. 18 subjects participated in one data collection session.

7. Experiments

The three classification algorithms were tested in five different experimental settings.

Base Case: This experiment was designed to test the general effectiveness of each classification technique. The testing and training data are taken from the same subject during the same collection session. The data from a collection session is divided into 28 training examples (seven examples per class) and 28 test examples (seven examples per class). Each example has a length of 512 samples.

Across People: This experiment was designed to to measure the average similarity between subjects. We take the data from one subject and produce 56 (14 examples per class) training examples of length 512 samples. We then attempt to classify 56 examples taken from a different subject, using a classifier trained on the data from the first. We perform this experiment pairwise on all subjects and then average the results.

We extended this experiment to see if training on multiple people would improve classification accuracy on a new person. We create five different classifiers: the first is trained on the data from one subject (as above), the second is trained on the data from two subjects, and so on. We then compare the performance of the classifiers on test data from a new subject.

Across Sessions This experiment is designed to measure the average similarity between data collected from the same subject across different sessions. We take the data from one session and produce 56 (14 examples per class) training examples of length 512 samples. We then attempt to classify 56 examples taken from data collected during a different session with the same subject, using a classifier trained on the data from the first. The experiment is performed on every pair of sessions, and then averaged.

We extended this experiment to see if training on the data from a single subject in multiple previous sessions would improve classification accuracy on the last one. We create five different classifiers: the first is trained on the data from one previous session (as above), the second is trained on the data from two previous sessions, and so on. We then compare the performance of the classifiers on the test from the final session.

We also performed experiments to compare various retraining techniques for the system. For the adaptation experiment, we calculate AR coefficients from 56 training examples (14 per class) collected from one session, and then attempt to classify 56 test examples collected from a different session.

Then, four signals (one for each class) are taken from the test-session data, and used to update the AR coefficients, using five different methods:

No Adaptation Weights are not changed.

Completely Retrain The AR coefficients are completely recalculated using the original training data and new examples.

New Data Retrain The AR coefficients are recalculated using only the new data given, and none of the original training data.

Adaptation The AR coefficients are updated using the Widrow-Hoff learning rule.

Conditional Adaptation The AR coefficients are updated using the Widrow-Hoff learning rule only if the classifier misclassified the new examples.

New Data Adaptation The AR coefficients are initially set by training on the new data for the first iteration, and then are adapted using Widrow-Hoff on the remaining new data.

After the update, then the remaining test signals are classified, and the procedure is repeated for 8 iterations and the results are plotted.

The final set of experiments compared the performance of HMM sequence decoding of EMG signals and Doerschuck's model for sequence decoding. Both the HMM model and Doerschuck's model were used to calculate the most probable state sequence for the EMG signal of a sequence of four different gestures. The results from both models are then compared to the actual state sequence and the error is calculated.

8. Results

8.1 Static Classification

The experimental results are summarized in Table 1. Mean values that are not significantly different from the best in each experimental condition are shown in bold.

The results for the base-case experiment were fairly good.

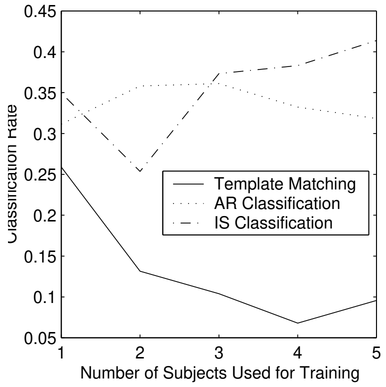

Figure 2. Classification rate vs. number of people represented in training data.

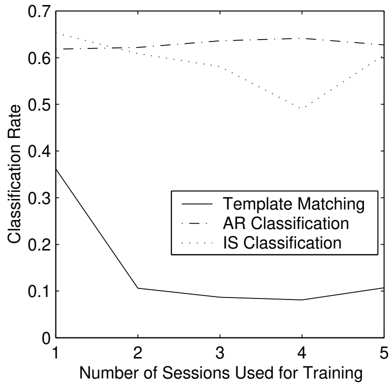

Figure 3. Classification rate vs. number of people represented in training data.

Table 1. Mean and standard deviations of classification accuracies for static classification methods in three main experimental conditions.

	NN	AR	IS
base case	40.1 (10.3)	**73.3** (20.5)	**77.8** (11.2)
across people	25.7 (6.9)	**33.2** (14.1)	**30.9** (13.1)
across sessions	35.8 (9.1)	**58.2** (15.1)	**51.0** (16.1)

As expected, nearest-neighbor on the raw data performed significantly worse than the other two methods. Although the mean performance of IS appears to be better than that of AR, the difference is not significant. We expect that the difference will be proved significant with larger sample sizes.

The across-people experiments show that there is very little similarity between the EMG signals of various people. Though certain groups of people are similar to each other, on average the signals differ greatly from person to person. Again, AR and IS perform significantly better than NN, but performance is poor everywhere. Results of the extended experiments, in which we trained on more than one person, are shown in Fig. 2. There was no improvement in classification rate.

The across-days experiments also show that there is considerable variation across days, even for a single subject, though it is less than the variation between subjects. Furthermore, the results of the extended experiments, shown in Fig. 3, demonstrate that training on more previous sessions does not improve performance.

8.2 Adaptation

The results obtained for the adaptation experiments are summarized in Fig. 4. They indicate that using new data only is more effective than any of the methods that incorporate the old data. This means that the signals change enough between days of use that weighting the data from the previous day equally with data from the current day actually hinders performance. However, if we were to incorporate previous data with a decreased importance weight, it might increase performance. Furthermore, the Widrow-Hoff algorithm performs just as well as batch retraining on the new data only. We thus can get all the speed and memory advantages of the Widrow-Hoff algorithm along with the performance of the batch algorithm.

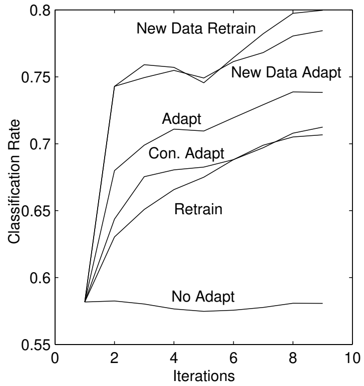

Figure 4. Performance of various adaptation schemes.

8.3 Sequence Decoding

The sequence decoding experiments show the HMM model outperforming Doerschuck's probability model on every test case. The average error rate for the HMM model was 84.49% (8.29%), while the error rate for the probability model was 72.85% (7.94%); the difference in the means is significant. Fig. 5 shows an example signal, true state sequence, output of the HMM model (labeled "viterbi") on that sequence, and output of the probability model on the sequence. The HMM sequence recognizes gesture changes with less noise and less latency than the probability model.

The classification results from the HMM are better than the best static results, demonstrating that the sequential information really does improve classification. The computational expense of the Viterbi method might prove problem-

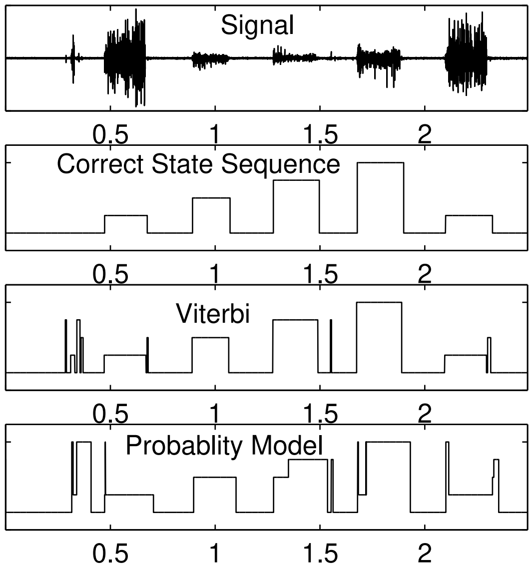

Figure 5. Comparison of sequence decoding with an HMM and a stateless model.

atic for very small portable devices, however.

9. Conclusions and Future Work

Although the 85% classification rate achieved with the AR + HMM technique is reasonably good, it is not yet sufficiently accurate for use in a consumer device, which requires essentially 100% accuracy.

There are a number of methods that might considerably improve the accuracy of the system, however. Many of them are improvements to the iterative scaling algorithm. It might be possible to make a "windowed" version of IS, that applies the Fourier transform to a sliding window of the signal and outputs a continuous stream of class probability estimates. These could then be incorporated into the HMM model to analyze the history at a higher level of abstraction. In addition, the current features used in IS analyze the data channels independently. Adding features that are sensitive to the cross-correlation between the channels could result in a significant improvement. Other ideas for improvement include using error correcting codes (Dietterich & Bakiri, 1995) and training the HMM transition weights from actual gesture sequences in the context of a particular application.

Extensions of the classification techniques explored here, together with more sophisticated signal sensing, may contribute to an important new user-interface modality.

Acknowledgements

We thank Will Hill for suggesting this research and for numerous helpful discussions. Thanks also to Anna Gilbert for valuable discussions.

References

Baum, L. E., & Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, *73*, 360–363.

Blum, A. (1997). Empirical support for winnow and weighted-majority based algorithms: results on a calendar scheduling domain. *Machine Learning*, *26*, 5–23.

Chen, S., Donoho, D., & Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, *20*, 33–61.

Coifman, R., & Wickerhauser, M. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, *39*, 713–718.

Csiszar, I., & Tusnady, G. (1984). Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplement Issue*, *1*, 205–237.

Della Pietra, S., Della Pietra, V., & Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*, 179–190.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Doerschuk, P., Gustafson, D., & Willsky, A. (1983). Upper extremity limb function discrimination using EMG signal analysis. *IEEE Transactions on Biological Engineering*, *30*, 18–29.

Englehart, K., Hudgins, B., Parker, P., & Stevenson, M. (1999). Classification of the myoelectric signal using time-frequency based representations. *Medical Engineering and Physics*, *21*, 431–438.

Freund, Y., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1997). Using and combining predictors that specialize. *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing* (pp. 334–343).

Graupe, D., & Cline, W. (1975). Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes. *IEEE Transactions on Systems Man and Cybernetics*, *5*, 252–259.

Kang, W., Shiu, J., nd J.S. Lai, C. C., Tsao, H., & Kuo, T. (1995). The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition. *IEEE Transactions on Biological Engineering*, *42*, 777–785.

Mallat, S., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, *41*, 3397–3415.

Porat, B. (1997). *A course in digital signal processing*. Wiley.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*, 1–40.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*, 96–104. Reprinted in *Neurocomputing* (MIT Press, 1988).