A MAXIMUM MARGIN DYNAMIC MODEL WITH ITS APPLICATION TO BRAIN SIGNAL ANALYSIS

XU WENJIE

(M. Eng., USTC, PRC)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Wu Jiankang, for his valuable advises from the global direction to the implementation details. His knowledge, kindness, patience, open mindedness, and vision have provided me with lifetime benefits. I am indebted to Dr. Wu for priceless and copious advice about selecting interesting problems, making progress on difficult ones, pushing ideas to their full development, writing and presenting results in an engaging manner.

I am grateful to Dr. Huang Zhiyong for his dedicated supervision, for always encouraging me and giving me many lively discussions I had with him. Without his guidance the completion of this thesis could not have been possible.

I'd also like to extend my thanks to all my colleagues in the Institute for Infocomm Research for their generous assistance and precious suggestions on getting over difficulties I encountered on the process of my research.

Many thanks to my friends who have had nothing to do with work in this thesis, but worked hard to keep my relative sanity throughout. I will not list all of you here, but my gratitude to you is immense. Lastly, but most importantly, my deepest gratitude to my parents, for their endless love, unbending support and constant encouragement. I dedicate this thesis to them.

Contents

Summary

The work in this dissertation is motivated by the application of Brain Computer Interface (BCI). Recent advances in computer hardware and signal processing have made it feasible to use human EEG signals or "brain waves" to communicate with a computer. Locked-in patients now have a means to communicate with the outside world. Even with modern advances, such systems still suffer from the lack of reliable feature extraction algorithm and the ignorance of temporal structures of brain signals. This is specially true for asynchronous brain computer interfaces where no onset signal is given. We have concentrated our research on the analysis of continuous brain signals which is critical for the realization of asynchronous brain computer interface, with emphasis on the applications to motor imagery BCI.

Having considered that the learning algorithms in Hidden Markov Model (HMM) does not adequately address the arbitrary distribution in brain EEG signal, while Support Vector Machine (SVM) does not capture temporary structures, we have proposed a unified framework for temporal signal classification based on graphical models, which is referred to as Kernel-based Hidden Markov Model (KHMM). A hidden Markov model was presented to model interactions between the states of signals and a maximum margin principle was used to learn the model. We

presented a formulation for the structured maximum margin learning, taking advantage of the Markov random field representation of the conditional distribution. As a nonparametric learning algorithm, our dynamic model has hence no need of prior knowledge of signal distribution.

The computation bottleneck of the learning of models was solved by an efficient two-step learning algorithm which alternatively estimates the parameters of the designed model and the most possible state sequences, until convergence. The proof of convergence of this algorithm was given in this thesis. Furthermore, a set of the compact formulations equivalent to the dual problem of our proposed framework which dramatically reduces the exponentially large optimization problem to polynomial size was derived, and an efficient algorithm based on these compact formulations was developed.

We then applied the kernel based hidden Markov model to the application of continuous motor imagery BCI system. An optimal temporal filter was used to remove irrelevant signal and noise. To adapt the position variation, we subsequently extract key features from spatial patterns of EEG signal. In our framework a mathematical process to combine Common Spatial Pattern (CSP) feature extraction method with Principal Component Analysis (PCA) method is developed. The extracted features are then used to train the SVMs, HMMs and our proposed KHMM framework. We have showed that our models significantly outperform other approaches.

As a generic time series signal analysis tool, KHMM can be applied to other applications.

List of Tables

List of Figures

Chapter 1

Introduction

With the significant enhancement of machine computation power in recent years, in machine learning community there is a rapid growing interest in modeling and analysis of the brain activities through capturing the salient properties of the brain signals, as for example eletroencephalography (EEG). The techniques are not only useful in a wide spectrum of brain signal related application areas including epilepsy detection, sleep monitoring, biofeedback and brain computer interfaces, but also in other application with complex time varying signals.

The work in this dissertation is motivated by the challenges we encountered in the Brain Computer Interface (BCI). One of such challenges is the lack of analysis algorithm which effectively address the temporal structures and complex distribution of brain signals. This is specially true for asynchronous brain computer interfaces where no onset signal is given. We have concentrated our research on the analysis of continuous brain signals which is critical for the realization of asynchronous brain computer interface, with emphasis on the applications to motor imagery BCI.

Having considered that the learning algorithms in Hidden Markov Model (HMM) does not adequately address the arbitrary distribution in brain EEG signal, while Support Vector Machine (SVM) does not capture temporary structures, we have proposed a unified framework for temporal signal classification based on graphical models, which is referred to as Kernel-based Hidden Markov Model (KHMM). A hidden Markov model was presented to model interactions between the states of signals and a maximum margin principle was used to learn the model. We presented a formulation for the structured maximum margin learning, taking advantage of the Markov random field representation of the conditional distribution. As a nonparametric learning algorithm, our dynamic model has hence no need of prior knowledge of signal distribution.

The computation bottleneck of the learning of models was solved by an efficient two-step learning algorithm which alternatively estimates the parameters of the designed model and the most possible state sequences, until convergence. The proof of convergence of this algorithm was given in this thesis. Furthermore, a set of the compact formulations equivalent to the dual problem of our proposed framework which dramatically reduces the exponentially large optimization problem to polynomial size was derived, and an efficient algorithm based on these compact formulations was developed.

We then applied the kernel based hidden Markov model to the application of continuous motor imagery BCI system. An optimal temporal filter was used to remove irrelevant signal and noise. To adapt the position variation, we subsequently extract key features from spatial patterns of EEG signal. In our framework a mathematical process to combine Common Spatial Pattern (CSP) feature extraction

method with Principal Component Analysis (PCA) method is developed. The extracted features are then used to train the SVMs, HMMs and our proposed KHMM framework. We have showed that our models significantly outperform other approaches. As a generic time series signal analysis tool, KHMM can be applied to other applications.

Because our work addresses the issues of time varying signal analysis in the brain computer interface, the following sections, we will start with concepts and research issues of brain computer interface, then come to the problem statement, and finally arrive at our contributions.

1.1 Brain Computer Interface

A brain-computer interface (BCI) is a *communication* system that does not depend on the brain's normal output pathways of peripheral nerves and muscles[RBH$^+$00]. Over the past fifteen years, the volume and pace of BCI research have grown rapidly. Encouraged by growing recognition of the needs and potentials of people with disabilities, new understanding of brain function, and the advent of powerful, low-cost computers, researchers have concentrated on developing new communication and control technology for people with severe motor disorders, such as amyotrophic lateral sclerosis (ALS), brainstem stroke, cerebral palsy, and spinal cord injury[Vau03].

The channels in the BCIs may be eletroencephalography (EEG), magnetroencephalography (MEG), positron emission tomography (PET), and functional magnetic resonance imaging (fMRI), which are available to monitor brain function. However, PET, fMRI and MEG are technically demanding and expensive. At

present, only EEG and related methods, which have relatively short time constants, can function in most environments, and require relatively simple and inexpensive equipment, offer the possibility of a new non-muscular communication and control channel, a practical BCI[WBM+02].

Since first described by Hans Berger in 1929, the EEG has been used mainly to evaluate neurological disorders in the clinic and to investigate brain function in the laboratory. Over that time, people have speculated that it might be used for communication and control, that it might allow the brain to act on the environment without the normal intermediaries of peripheral nerves and muscles. However, this idea attracted little serious research activities but some popular scientific fiction authors until recently, for at least 3 reasons[WBM+02].

1. The resolution and reliability of the information detectable in the spontaneous EEG is limited by the vast number of electrically active neuronal elements, the complex electrical and spatial geometry of the brain and head, and the disconcerting trial-to-trial variability of brain function.

2. EEG-based communication requires the capacity to analyze the EEG in real-time, and until recently the requisite technology either did not exist or was extremely expensive.

3. There was in the past little interest in the limited communication capacity that a first–generation EEG-based BCI was likely to offer.

Like any communication or control system, a BCI has input (e.g. electrophysiological activity from the user), output (e.g. device commands), components that translate input into output, and a protocol that determines the onset, offset, and

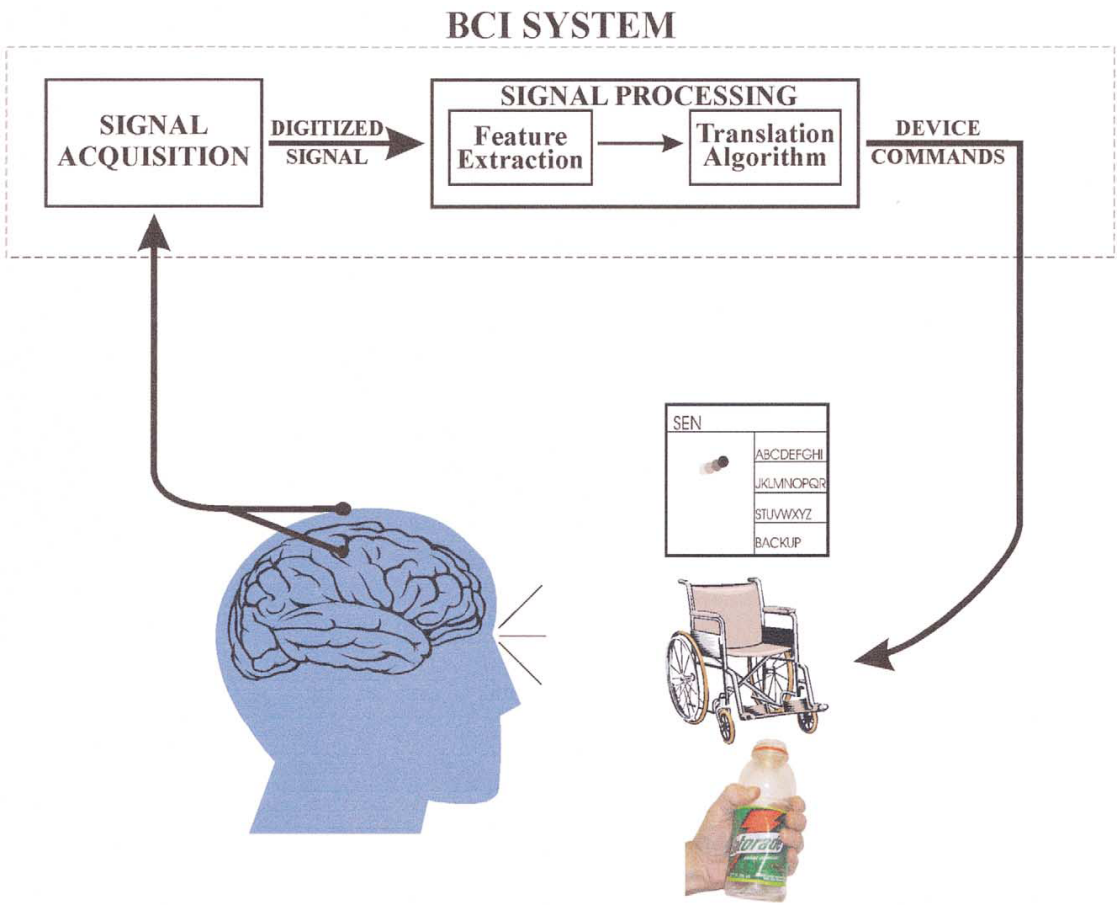


Figure 1.1: Signals from the brain are acquired by electrodes on the scalp or in the head and processed to extract specific signal features that reflect the user's intent. These features are translated into commands that operate a device. Success depends on the interaction of two adaptive controllers, user and system.

timing of operation (Figure 1.1). The key components in a BCI system are signal acquisition, feature extraction and translation algorithm, which decide the performance of the system measured by speed and accuracy.

- Signal acquisition

  While implanted EEG electrodes can be used to monitor the brain activities

that drive a cursor on a computer monitor [KBM+00], the non-invasive methods is providing to be viable and is obviously preferable. These approaches can be broadly categorized as visual evocation [Sut92, MMCJ00], P300 evocation [DSW00], operant conditioning[BGH+99] and cognitive tasks[PN01].

The former two approaches rely on the visual evoked potentials or the P300 evoked potentials, which are generated by some visual stimuli. They usually require a structured environment and mostly just provide the user with the ability to choose from a set of options.

Like the previous two, the operant conditioning rely on biofeedback to allow the subject to acquire the automatic skill of controlling EEG signals in order to move the cursor or make a selection. But it requires initial user training. Over many training sessions the subject acquires the skill of controlling the movement of the cursor without being consciously aware of how this is achieved. This approach may be compared to the skill of riding a bicycle or playing tennis, where employment of the skill is voluntary but automatic.

The BCI systems with cognitive or mental tasks can be deemed the second–generation of BCI. Unlike with operant conditioning, the subjects perform specific thinking tasks. Cognitive tasks are asynchronous and do not need any biofeedback procedure, which suggests that it could be good communication channels of the BCI systems.

So far, the cognitive task most commonly used in BCI studies is motor imagery, as it produces changes in EEG that occur naturally in movement planning and are relatively straightforward to detect. With appropriate feature extraction algorithm and classifier, the maximum information transfer rate

is possible reached up to 24 bits/min[PN01]. However, motor imagery tasks may be inappropriate for certain groups of subjects who have been paralyzed for many years, or indeed from birth.

- Feature extraction

  The performance of a BCI, like that of other communication systems, depends on its signal-to-noise ratio (SNR). The goal is to recognize and execute the user's intent, and the signals are those aspects of the recorded electrophysiological activity that correlate with and thereby reveal that intent. This correlation can be maximized by employing feature extraction methods which are to greatly affect SNR, without consideration of the impact of the user. To achieve this goal, consideration of the major sources of noise is essential. No good performance can be reached without enhancing the signal and reducing the noise from:

  - Nonneural sources. These include other human's activity (e.g. muscle activation and eye movements) and interference (e.g. 60-Hz line noise).
  - Neural sources. These are the EEG features that come from central nervous system (CNS) other than those used for communication.

  Noise resulting from interference can, to a certain degree, be prevented by conducting the data acquisition in a controlled environment, e.g. keeping the human subject and recording apparatus as remote as possible from the electrical supply and electrically powered equipment, shielding from electrostatic interference, and avoiding magnetic induction by disallowing loops of significant area in current-carrying leads. In addition to this, some noise the

radio frequency interference can be filtered out at the inputs of recording amplifiers since the signals of interest exist in a narrow low frequency band. Noise detection and discrimination problems are greatest when the characteristics of the noise are similar in frequency, time or amplitude to those of the desired signal. For example, eye movements are of greater concern than EMG when a slow cortical potential is the BCI input feature because EOG and SCP have overlapping frequency ranges. For the same reason, EMG is of greater concern than EOG when a β rhythm is the input feature. Therefore, how to design the feature extraction algorithm strongly depends on the specific signal used in the BCI system.

A variety of options for improving BCI signal-to-noise ratios are under study. These including spatial and temporal filtering techniques, signal averaging, and single-trial recognition methods. Much work up to now has focused on showing by offline data analyses that a given method will work. Although strong in minimizing or removing non-CNS artifacts, these methods might be inappropriate to CNS activities. This is because:

- The concurrency of brain activities is little of concern. These methods thought that all the signals for offline analysis or online translation come from the same underline brain function so that they bring many uncorrelated signals or noise to the classifier and make the wrong decision.
- The underline brain function or neural activity is litter of concern. These methods consider the brain that generates the interested signal as the *blackbox*

- Classification

  As mentioned before, a BCI system is not designed to understand all the mind users is thinking, but to train the users to provide some defined brain signals and decide what the signals are. From pattern recognition view, this system is to provide a decision rule which decides which category the signal belongs to. To reach this goal well, the approach employed in BCI systems have to match the critical features of brain signals.

  So far, we do not have a clear understanding of the brain and how the brain makes brain signals. This situation is much worse when the brain signals correspond to the activities in populations of neurons. Therefore, knowledge-driven classification approaches are not appropriate to the non-invasive BCI systems. On the contrary they incline to use data-driven methods. Compared to knowledge-drive approaches, these methods do not need or need less prior knowledge while directly learn the decision rules (knowledge) from the labeled/unlabeled samples.

  The discriminant approaches, as an important class of data-driven methods, are heavily used in conventional BCI system. They attempt to classify samples by constructing hyperplanes, which are estimated from the training samples. These samples are assumed have a underlying class conditioned set of probabilities and/or probabilities density functions. Interestingly, these methods have discrimination capability between classes and thus can promise better performance.

  Previous analyses of EEG signals attested that only the EEG signals within a short length, usually less than $1s$, can be deemed to be stationary signals.

In the case of asynchronous BCI, the input brain signals would be the continuous signals so that the temporal structures of the EEG signals can not be ignored. Therefore, it violates the assumption of the discriminant approaches and may degrade the performance of the BCI systems using the discriminant approaches.

In short, numerous concurrent brain activities and interfering noises make the BCI problem much more intricate. Achievements in technologies of BCI have little effort to make the brain computer interface applications go out of the lab. It may due to a lack of reliable feature extraction algorithm and the ignorance of temporal structures of brain signals. In this thesis we shall address these BCI issues and propose possible solutions.

1.2 Problem statement

The challenging issue that we are addressing is asynchronous brain computer interfaces where no onset signal is given. We concentrate our research on the analysis of continuous brain signals which is critical for the realization of asynchronous brain computer interface, with emphasis on the applications to motor imagery BCI. We do not address the classification problems of other types of temporal signals. However, some of our research results are actually applicable to those real temporal signals, for example speech signals.

We further state the issues as follows:

- Propose a dynamic model for the brain signal classification. Modeling the

temporal structure is inevitable if the onset timing is unknown in the asynchronous BCI systems. Furthermore, the emphasis on dynamics help us enhance a brain signal corrupted by noise and transmission distortion and realize the practical BCI systems in a very efficient manner. In summary, dynamic model is one of major building blocks for building high performance BCI systems.

- design the reliable feature extraction methods to maximize the correlation between the user's intent and the recorded brain signal. In our research, the brain signal is recorded on a multitude of channels placed in a dense grid covering large parts of the brain. Given that a brain activity originate from very localized areas in the cortex, we expect that not all signals recorded from different sites contribute the same amount of information to the classification, and some may only contribute noise. Furthermore, appropriate temporal filtering can also enhance signal-to-noise ratios. Usually, only specific narrow spectral bands of the brain signal are relevant to the user's intend we want to decipher. Designing of the reliable feature extraction methods is hence vital to build an high performance brain computer interfaces.

- Develop an integrated BCI system framework which provides ready solutions to applications to help lock-in people freely communicate with outsides. It includes system modeling, the individual brain activities connecting strategy, and the reject mechanism for undesired brain activities, etc.

1.3 Contribution of the thesis

This thesis addresses the problem of efficient learning of high-accuracy models for human-computer communication problems. Having studied the whole BCI system, including the brain signal's creation, processing, and translation in this system, we have designed a system framework with respect to the technical aspect of brain computer interfaces. Three key issues have been identified and novel methods have been developed as solutions to the three issues:

1. A kernel based hidden Markov model for temporal signal prediction problem. We have proposed a unified framework for temporal signal classification based on graphical models. A hidden Markov model is presented to model interactions between the states of signals. An alternative to likelihood-based methods, this framework builds upon the large margin estimation principle. Intuitively, we find parameters such that inference in the model (dynamic programming, combinatorial optimization) predicts the correct answers on the training data with maximum confidence. We develop general conditions under which exact large margin estimation is tractable and present a formulation for the structured maximum margin learning, taking advantage of the Markov random field representation of the conditional distribution. As a nonparametric learning algorithm, our dynamic model has hence no need of prior knowledge of signal distribution while providing a strong generalization mechanism.

2. A two-step learning algorithm for solving the training problem of the kernel

based hidden Markov model. We have developed an efficient two-step learning algorithm for solving the training problem of the kernel based hidden Markov model. Due to a complete absence of the labels of states in most of cases of temporal signal classification, we have to face the chief computational bottleneck in learning the parameters of models. The two-step learning algorithm solved this problem by alternatively estimating the parameters of the designed model and the most possible state sequences, until convergence. The proof of convergence of this algorithm was given in this thesis. Furthermore, a set of the compact formulations equivalent to the dual problem of our proposed framework which dramatically reduces the exponentially large optimization problem to polynomial size is derived, and an efficient algorithm based on these compact formulations was developed.

3. A motor imagery BCI framework based on the KHMM We have developed a continuous BCI system which just requires the user imagining his/her hand movement. Our framework was built on the basis of our proposed kernel based hidden Markov model which has a good generalization property and gives a minimum empirical risk. Specifically, an optimal temporal filter was employed to remove irrelevant signal and subsequently extract key features from spatial patterns of EEG signal which transforms the original EEG signal into a spatial pattern and applies the RBF feature selection method to generate robust feature. All the extracted features were then classified by the left and right hand imagine models trained using the two-step learning algorithm. Our experimental results have shown significant improvement in classification accuracy over SVMs and HMMs.

1.4 Overview of the thesis

We discuss related works on BCI system architectures in Chapter 2. In Chapter 3, we proposed the kernel based hidden Markov model for temporal signal classification problem, followed by an efficient learning algorithm in chapter 4. Chapter 5 discusses a continuous motor imagery BCI system based on kernel based hidden Markov framework. The thesis is concluded in Chapter 6.

Chapter 2

Background

Can these observable electrical brain signals be put to work as carriers of information in man-computer communication or for the purpose of controlling such external apparatus as prosthetic devices or spaceships? Even on the sole basis of the present states of the art of computer science and neurophysiology, one may suggest that such a feat is potentially around the corner. - Vidal [Vid73]

In 1973, Jacques Vidal published an article on the first BCI. In the 23-page paper, most of the space was devoted to describing EEG signal acquisition hardware/software and the signal processing of the obtained EEG signals. Real-time acquisition is imperative for a BCI system and the existing computer equipment was not up to the task. Still, many of the concepts used today in BCIs were discussed in Vidal's paper. After describing the future possibilities for BCIs, Vidal talked about neurophysical considerations. What brain signals should be used for a BCI and what were the properties of these signals? Vidal mentioned alpha rhythms, evoked potentials, and even event-related synchronization/desynchronization (ERS/ERD)

of the EEG, all of which are used in BCIs today. The idea for advanced processing of single trial evoked potentials using principal component analysis appeared in Vidal's paper as well as the more common spectral analysis of EEG signals. The goal of the paper was to indicate the necessary components for a working BCI and this was done very well. Even with its forward thinking, Vidal could not have foreseen some of the more modern issues associated with getting a BCI to work well. These BCI system issues include designing the user application while taking human factors into consideration as well as the overall BCI system architecture.

2.1 The Nature of the EEG and Some Unanswered Questions

Much is known and much remains a mystery about the nature of EEG signals. Knowledge about EEG signals may help the BCI researcher in two ways. First, knowledge may help the researcher choose what signal conveys the most information for control and second, it may aid in developing signal processing algorithms for detecting the relevant signal. Lack of knowledge hinders the BCI researcher. When the true nature of the signal is unknown, it is difficult to choose the most appropriate signal processing routine for recognition.

Traditionally, electroencephalogram (EEG) is a display of brain voltage potentials written onto paper over time. A modern system for EEG acquisition digitizes these potentials for computer storage, although systems that output directly onto paper remain in use. Electrodes passively conduct voltage potentials from columns of neurons in the brain and must pick up microvolt level signals. The signal to

noise ratio must be kept as high as possible and electrodes are constructed from such materials as gold and silver chloride in order to aid in this. Various conductive gels or pastes are used between an individual's skin and the electrode in order to reduce the impedance between the electrode and the scalp as much as possible.

Configurations of electrodes usually follow the International 10-20 system of placement [Jas58], although larger electrode arrays may follow the Modified Expanded 10-20 system as proposed by the American EEG Society (see Figure 2.1). The introduction of the Modified Expanded 10-20 system indicates an increase in the normal application of an expanded number of electrodes. Not surprisingly, more electrodes means increased spatial resolution of the signal over the head and arrays with as many as 256 electrodes have been used successfully in research applications.

The availability of large numbers of electrodes introduces the problem of how to connect them to the recording device. A plethora of different configurations exist, but two main classes of configurations or montages arise from the possibilities: referential and bipolar montages.

The distinguishing feature of referential montages is that all electrode potentials are calculated with respect to a reference electrode placed in an electrically quiet area. The main advantage of such a recording method is that referential recording can give an undistorted display of the shape of potential changes and is especially useful for the recording of potentials with a wide distribution. Since differential amplifiers are used, referential montages also make it simple to mathematically calculate other kinds of montages after recording.

Unfortunately, it is essentially impossible to find a reference electrode that is

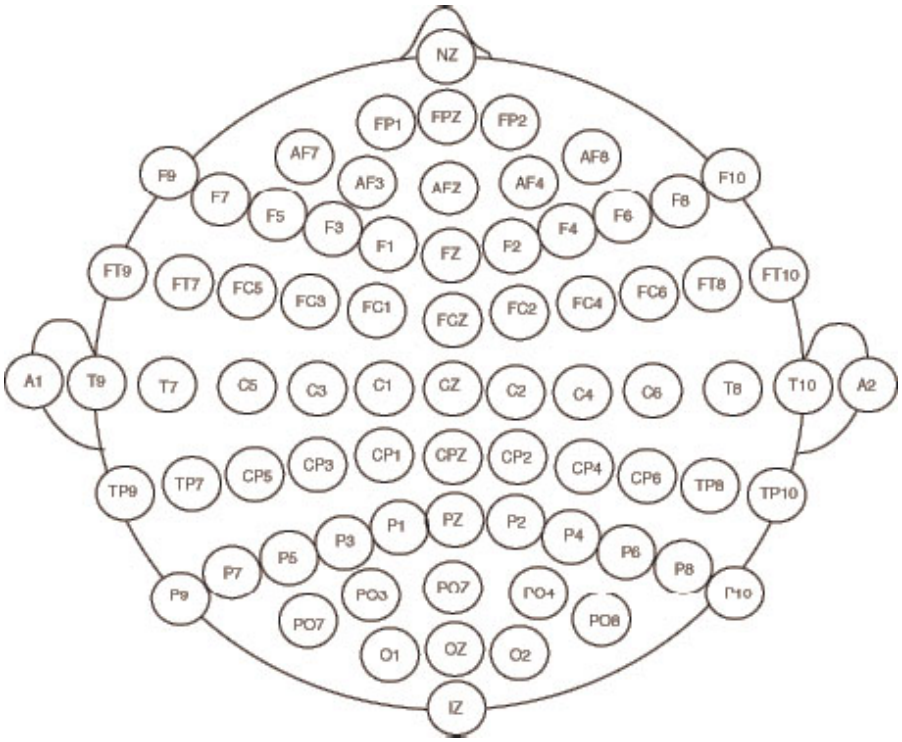


Figure 2.1: The extended 10-20 system for electrode placement. Even numbers indicate electrodes located on the right side of the head while odd numbers indicate electrodes on the left side. The letter before the number indicates the general area of the cortex the electrode is located above. A stands for auricular, C for central, Fp for prefrontal, F for frontal, P for parietal, O for Occipital, and T for temporal. In addition, electrodes for recording vertical and horizontal electro-oculographic (EOG) movements are also place. Vertical EOG electrodes are placed above and below an eye and horizontal EOG electrodes are placed on the side of both eyes away from the nose.

entirely inactive. Reference electrodes located everywhere from the ear to the big toe have failed in the attempt to find a truly quiet reference. In order to help overcome this problem, average reference electrodes (where two electrode sites contribute equally to the reference electrode) may be used. The most common average reference electrode configuration is known as the linked ears configuration due to the equal contribution of A1 and A2 to the reference electrode. A1 and A2 may also be attached to the mastoids instead of the ears, in which case the reference is known as a linked mastoid configuration. In order to remove the influence of the reference location from the recording, techniques such as the Hjorth transform [Hjo75] may be used.

Bipolar montages connect pairs of electrodes to the inputs of amplifiers. As an example, the longitudinal bipolar montage connects Fp1-F3, F3-C3, C3-P3, P3-O1, and so on, forming rows of electrodes. The advantage of these types of montages is that they distinguish local activity much more clearly than a referential montage. The disadvantage of bipolar montages is that they may distort the wave shape and amplitude of widely distributed potentials.

Clearly, the type of montage used will greatly affect the ability of a system to recognize certain events in the signal. Since BCIs tend to deal with widely distributed signals, most BCIs use a referential montage. After a montage is chosen, the electrode voltage potentials are differentially amplified on the order of ten to twenty thousand times the original voltage. As discussed in Spehlmann's EEG Primer [Spe91], the EEG reader needs to distinguish the following features: waveform, repetition, frequency, amplitude, distribution, phase relation, timing, persistence, and reactivity. These are common features distinguished by BCIs.

Waveforms may be regular, having a fairly uniform appearance due to symmetrical rising and falling phases. One example of a regular waveform would be a sinusoidal wave. Other waveforms may be irregular, having uneven shapes and durations. The waveform frequencies of particular interest to clinical EEG readers range from 0.1 Hz to around 20 Hz. Many frequencies are apparent in the normal EEG and frequency bands help to set apart the most normal and abnormal waves in the EEG, making frequency an important criteria for assessing abnormality in clinical EEG. As electrodes are positioned over different parts of the head, the electrical activity recorded may appear over large or small areas. This is the distribution of a wave. Distributions may be lateralized on one side of the head or may be diffuse. Focal activity is activity that is restricted to one or a few electrodes over an area of the head. The reactivity of a signal refers to changes that may be produced in some normal and abnormal patterns by various maneuvers. A common example of this is the blocking of the alpha rhythm by eye opening or other alerting procedures [Spe91].

While some descriptors of the EEG signal seem fairly obvious, there are others that have created controversy in the EEG community. One of the obvious questions on the nature of the EEG signal remains unknown - is the system linear or nonlinear? It is also unknown how chaotic the data is. Without the answers to these questions, it remains difficult to choose the proper routines for EEG signal recognition. Toda, Murai, and Usui present a measure of nonlinearity in time series [TMU92]. The measure of nonlinearity is calculated from the weights of a trained feedforward neural network with nonlinear hidden units. As examples, they measure the nonlinearity of sunspot series and a carp's EEG. The sunspot is (of

course) found to be nonlinear, but the carp's EEG is linear. While there are problems with this approach, such as the lack of complete data sets and noise effects, the approach raises the question of the possibility of globally linear neurocortical dynamics. Freeman's nonlinear model for the neocortex assumes chaotic nonlinear dynamics [Fre91, Fre95]. Pyramidal cells are important neurons in the neocortex and Freeman's model predicts that the shar nonlinearity of the neuronal threshold could cause chaotic dynamics if both the firing rate and the field potential of any pyramidal cell were raised above a critical level of excitation. Simulations of his principles have yielded the predicted chaotic dynamic properties.

There is no incontrovertible proof that the EEG reflects any simple chaotic process [WL96]. Fundamental difficulties lie in the applicability of estimation algorithms to EEG data, because of limitations in the size of data sets, noise contamination, and lack of signal stationarity. Even with locally chaotic dynamics, does this mean that there must be globally chaotic dynamics? An important class of simulation studies suggest this must be the case [Kan90, Kan92]. These studies concern one-dimensional chaotic numerical subprocesses of considerable generality (one-dimensional chaotic maps) that are globally coupled, each to all others. Such coupled maps exhibit global chaos and appear to escape from the law of large numbers and the central limit theorem. However, the escape from the law of large numbers does not occur in the presence of noise (a common element in any EEG) [Kan90, Kan92].

The nonlinear model proposed by Freeman contrasts with one proposed by Nunez [Nun95]. Nunez's model treats the EEG signal as a linear wave process

and the global dynamics of the brain are treated as a problem of the mass action of coupled neuronlike elements [WL96]. While Freeman's model predicts an oscillation caused by neuronal firing at around 40 Hz that is consistent with experimental findings, Nunez's model predicts a wave propagation velocity of 7-11 m/sec for human alpha waves that is also consistent with experimental findings. Either model appears consistent with some experimental data, but is either model correct? Interestingly enough, due to the noise in an EEG signal, both models could be correct. Freeman's model might actually agree with Nunez's globally linear model for neocortical EEG.

Since the nature of EEG signals is unknown, difficulties lie in trying to decide on a particular signal recognition routine. At best, if EEG signals are linear, then the linear recognition algorithms that most BCIs use may be sufficient. At worst, linear recognition algorithms are poor descriptors of the signals they hope to recognize.

2.2 Neurophysiological Signals Used in BCIs

What signals should be used for control in a BCI? This is an open question in the field and quite a few signals are in current use. As previously stated, signals may be broken into three general categories: implanted methods, evoked potentials, and operant conditioning. Both evoked potential and operant conditioning methods are normally externally-based BCIs as the electrodes are located on the scalp. Table 2.1 describes the different signals in common use. It may be noted that some of the described signals fit into multiple categories. As an example, single neural recordings may use operant conditioning in order to train neurons for control or

may accept the natural occurring signals for control. Where this occurs, the signal is described under the category that most distinguishes it.

Several questions are of relevance when considering what signal to use for a proposed BCI:

1. **What remaining control is necessary in order to use the BCI?**

   Some BCIs require the use of eye movement control and some do not require any remaining motor control.

2. **Does the user of the BCI need to be trained in order to elicit the necessary signal for control and if so, then how long does the training last?**

   Operant conditioning methods may require extensive training in order to use them for control.

3. **What percentage of the population can obtain control using the signal?**

   While almost everybody has apparent evoked potentials, not everybody appears to be able to use biofeedback in order to learn how to use a BCI based on operant conditioning. This is discussed further below.

4. **Does the signal provide continuous or discrete control?**

   Evoked potentials may only provide discrete control. Operant conditioned signals may provide continuous control, because they are obtained from ongoing EEG activity.

5. **Does the nature of the signal change over time?**

Signal Name	Description
Mu, and Alpha Rhythm Operant Conditioning	The mu rhythm is an 8-12 Hz spontaneous EEG rhythm associated with motor activities and maximally recorded over sensorimotor cortex. The alpha rhythm is in the same frequency band, but is recorded over occipital cortex. The amplitudes of these rhythms may be altered through biofeedback training.
Event-related Synchronization/Desynchronization (ERS/ERD) Operant Conditioning	Movement-related increases and decreases in specific frequency bands maximally located over sensorimotor cortex. Individuals may be trained through biofeedback to alter the amplitude of signals in the appropriate frequency bands. These signals exist even when the individual imagines moving as the movement-related signals are preparatory rather than actual.
Slow Cortical Potential Operant Conditioning	Large negative or positive shifts in the EEG signal lasting from 300ms up to several minutes. Individuals may be trained through biofeedback to produce these shifts.
P3 Component of the Evoked Potential	A positive shift in the EEG signal approximately 300-400ms after a task relevant stimulus. Maximally located over the central parietal region, this is an inherent response and no training is necessary.
Short-Latency Visual Evoked Potentials	To produce the component, a response to the presentation of a short visual stimulus is necessary. Maximally located over the occipital region, this is an inherent response and no training is necessary.
Individual Neuron Recordings	Individuals receive implanted electrodes that may obtain responses from local neurons or even encourage neural tissue to grow into the implant. Operant conditioning may be used to achieve control or the natural response of a cell or cells may be used.
Steady-State Visual Evoked Potential (SSVER)	A response to a visual stimulus modulated at a specific frequency. The SSVER is characterized by an increase in EEG activity at the stimulus frequency. Typically, the visual stimulus is generated using white fluorescent tubes modulated at around 13.25 Hz or by another kind of strobe light. A system may be constructed by conditioning individuals to modulate the amplitude of their response or by using multiple SSVERs for different system decisions.

Table 2.1: Common signals used in BCIs

Many of the signals currently used may change as a function of fatigue.

6. **Does the signal necessitate an invasive procedure in order to work?**

   While most BCIs obtain control using electrodes on the scalp, implanted methods are invasive.

Implanted methods use signals from single or small groups of neurons in order to control a BCI. These methods have the benefit of a much higher signal-to-noise ratio at the cost of being invasive. They require no remaining motor control and may provide either discrete or continuous control. Chapin and Gaal have successfully recorded up to 46 neurons and used their natural responses to enable four out of eight rats to obtain water with the neural processes [CG99, CMMN99]. While most systems are still in the experimental stage, Kennedy's group has forged ahead to provide control for locked-in patient JR [Kan90, Kan92]. Kennedy's approach involves encouraging the growth of neural tissue into the hollow tip of a two-wire electrode known as a neurotrophic electrode. The tip contains growth factors that spur brain tissue to grow through it. Through an amplifier and antennas positioned between the skull and the scalp, the neural signals are transmitted to a computer, which can then use the signals to drive a mouse cursor. This technique has provided stable long term recording and patient JR has learned to produce synthetic speech with the BCI over a period of more than 426 days. It is unknown how well this technique would work on multiple individuals, but it has worked on both patients (JR and MH) who have been implanted.

Evoked potentials (EPs) are usually obtained by averaging a number of brief EEG segments time-registered to a stimulus in a simple cognitive task. In a BCI,

EPs may provide control when the BCI application produces the appropriate stimuli. This paradigm has the benefit of requiring little to no training to use the BCI at the cost of having to make users wait for the relevant stimulus presentation. EPs offer discrete control for almost all users as EPs are an inherent response.

Exogenous components, or those components influenced primarily by physical stimulus properties, generally take place within the first 200 milliseconds after stimulus onset. These components include a Negative waveform around 100 ms (N1) and a Positive waveform around 200 ms after stimulus onset (P2). Visual evoked potentials (VEPs) fall into this category. Sutter uses short visual stimuli in order to determine what command an individual is looking at and therefore wants to pick [Sut92]. He also shows that implanting electrodes improves performance in an externally based BCI.

In a different approach, McMillan and colleagues have trained volunteers to control the amplitude of their steady-state VEPs to florescent tubes flashing at 13.25 Hz [JMCM98, MMCJ99, VWD96]. Using VEPs has the benefit of a quicker response than longer latency components. The VEP requires that the subject have good visual control in order to look at the appropriate stimulus and allows for discrete control. As the VEP is an exogenous component, it should be relatively stable over time.

Endogenous components, or those components influenced by cognitive factors, take place following the exogenous components. Around 1964, Chapman and Bragdon [CB64] as well as Sutton et. el. [SBZJ65] independently discovered a positive wave peaking at around 300 ms after task-relevant stimuli. This component is known as the P3 and is shown in Figure 3.1. While the P3 is evoked by many

types of paradigms, the most common factors that influence it are stimulus frequency (less frequent stimuli produce a larger response) and task relevance. The P3 has been shown to be fairly stable in locked-in patients, re-appearing even after severe brain stem injuries [OMTF96]. Farwell and Donchin first showed that this signal may be successfully used in a BCI [FD88]. Using a broad cognitive signal like the P3 has the benefit of enabling control through a variety of modalities, as the P3 enables discrete control in response to both auditory and visual stimuli. As it is a cognitive component, the P3 has been known to change in response to subject fatigue. In one study, a reduction in the P3 was attributed to fatigue after subjects performed the task for several hours [dSvLR86].

As shown in Table 2.1, several methods use operant conditioning on spontaneous EEG signals for BCI control. The main feature of this kind of operant conditioning is that it enables continuous rather than discrete control. This feature may also serve as a drawback: continuous control is fatiguing for patients and fatigue may cause changes in performance since control is learned. As shown by the various groups using these methods, operant conditioning methods using spontaneous EEG are not easily learned by everybody.

Wolpaw and his colleagues train individuals to control their mu rhythm amplitude (discussed in Table 2.1) for cursor control [WMNF91]. Mu rhythm control does not require subjects to have any remaining motor control. For the cursor control task, normal subjects are trained on the order of 10-15 sessions in order learn to move the cursor up/down. In the several papers examined, it appears that not all subjects obtain control, although most seem to during this time frame. It is normal to see four out of five subjects who obtain greater than 90% accuracy with

the other one obtaining around chance [WMNF91]. This implies that somewhere around 80% of the subjects may obtain good control.

In related work, the Graz brain-computer interface trains people to control the amplitude of their ERS/ERD patterns. Subjects are trained over a few sessions in order to learn a cursor control task. As in the mu rhythm control, not all subjects learn to control the cursor accurately. Obtaining two out of six subjects who are not able to perform the cursor control task has been reported [PKN+96]. Part of the charm of this system is that it gives biofeedback to the user in the form of a moving cursor after training. The use of areas over the sensorimotor cortex for both ERS/ERD and mu rhythm control might pose a problem in people with ALS because the cortical Betz cells in the motor cortex may die in the later stages of the disease [BS96].

Slow cortical potentials serve as the signal in the Thought Translation Device, a communication device for ALS patients created by Birbaumer's group in Austria [BGH+99]. Since this system is used with patients, it is difficult to tell how hard it is to learn the system. Patients may be medicated, depressed, or fatigued: all of which affect learning rates. Subjects are trained over several months to use the system. All subjects that have wanted to learn the system seem to have been successful. No remaining motor control is necessary in order to use the Thought Translation Device. Unlike mu rhythm control or ERS/ERD, the slow cortical potential has not been used for continuous control. It may take many seconds in order to produce and hold a slow cortical potential in order to trigger the system.

While the signals discussed are used currently, other signals may be possible. Several papers have been written on recognizing EEG signal differences during

different mental calculations. These papers suggest that different parts of the brain are active during different types of mental calculation, and if these different tasks may be accurately recognized, they could be used in a BCI. Lin et. al. [LTL93] describe a study where five tasks were compared: multiplication problem solving, geometric figure rotation, mental letter composing, visual counting, and a baseline task where the subject was instructed to think about nothing in particular. Results from this experiment suggest that the easiest tasks to identify are multiplication problem solving and geometric figure rotation, but even these tasks are not easily identified. Other papers have concentrated on mental tasks, but none have found easily recognizable differences between different tasks [Dev96, FHR+95].

2.3 Existing Systems

Current systems range from simple experimental interfaces meant to test the suitability of a specific EEG signal to full applications used by patients. The system includes the hardware used in the BCI, the underlying BCI backend software, and the user application. While the hardware used in a research testbed does not matter as long as it performs as needed, expense, portability, and reliability become very real issues in a BCI for patient use.

The underlying BCI backend software is not discussed in many papers. It is, however, as important as the hardware. The backend includes software for reading in the EEG signals, scheduling them for processing, and processing them into a form that may be used by the user application. The backend software determines the BCI portability, extendibility, and flexibility. It also determines how maintainable the software will be over a period of time. For instance, the construction of the

software may provide the flexibility to enable users to choose from a wide variety of user applications or the user may only be able to use one application if the BCI system is monolithic.

In assessing current user applications, it is important to consider the usability of the application. The field of human factors tells us repeatedly that a poorly designed user application may injure performance. This applies to a BCI as well as to many other items in everyday use and will occur regardless of the signal recognition routines used. Several important factors should be considered in the design of the application, including the following five mentioned by Ben Shneiderman [Shn98]:

1. What is the time to learn the system?
2. What is the speed of performance?
3. How many and what kinds of errors do users make?
4. How well do users maintain their knowledge after an hour, a day, or a week? What is their retention?
5. How much did users like using various aspects of the system? What is their subjective satisfaction?

Several features of existing BCIs are compared in Table 2.2. Surprisingly, most BCI papers do not discuss subjective satisfaction at all and so the category for subjective satisfaction only includes whether or not it was considered in the papers about the system. In addition to these considerations, the application designer might want to consider the following general goals as specified by the U.S. Military Standard for Human Engineering Design Criteria [Shn98]:

1. Achieve required performance by operator, control, and maintenance personnel
2. Minimize skill and personnel requirements and training time
3. Achieve required reliability of personnel-equipment combinations
4. Foster design standardization within and among systems

When measured using these considerations, all BCIs fall short in some manner. This could be because most BCIs are research instruments or grow out of a research project. In the future, it will be very important to consider the system-wide aspects of BCIs.

2.3.1 The Brain Response Interface

Sutter's Brain Response Interface (BRI) [Sut92] is a system that takes advantage of the fact that large chunks of the visual system are devoted to processing information from the foveal region. The BRI uses visually evoked potentials (VEP's) produced in response to brief visual stimuli. These EP's are then used to give a discrete command to pick a certain part of a computer screen. This system is one of the few that have been tested on severely handicapped individuals. Word processing output approaches 10-12 words/min. and accuracy approaches 90% with the use of epidural electrodes. This is the only system mentioned that uses implanted electrodes to obtain a larger, less contaminated signal.

A BRI user watches a computer screen with a grid of 64 symbols (some of which lead to other pages of symbols) and concentrates on the chosen symbol. A specific subgroup of these symbols undergoes a equiluminant red/green fine check

System	Training Time	Number of Choices	Speed	Errors	Retention	Subjective Satisfaction
Brain Response Interface 2.3.1	10-60 minutes	64	30	10%	Excellent	Considered
SSVEP Training [MMCJ00]	6 hrs.	N/A	N/A	20% or less	Not mentioned	Not Discussed
P3 Character Recognition [FD88]	Minutes	36	4	5%	Excellent	Not Discussed
Mu Rhythm Training [WMNF91]	15-20 sessions	2	20	10%	Not mentioned	Not Discussed
ERS/ERD 2.3.3	2-2.5 hrs.	2	N/A	11% or less	Not mentioned	Not Discussed
Thought Translation Device [BGH+99]	Months	27	2	10-30%	Not Good	Indirectly discussed
Implanted Device	Months	N/A	2	Not reported	Excellent	Considered

Table 2.2: A comparison of several features in existing BCIs

or plain color pattern alteration in a simultaneous stimulator scheme at the monitor vertical refresh rate (40-70 frames/s). Sutter considered the usability of the system over time and since color alteration between red and green was almost as effective as having the monitor flicker, he chose to use the color alteration because it was shown to be much less fatiguing for users. The EEG response to this stimulus is digitized and stored. Each symbol is included in several different subgroups and the subgroups are presented several times. The average EEG response for each subgroup is computed and compared to a previously saved VEP template (obtained in an initial training session), yielding a high accuracy system.

This system is basically the EEG version of an eye movement recognition system and contains similar problems because it assumes that the subject is always looking at a command on the computer screen. On the positive side, this system has one of the best recognition rates of current systems and may be used by individuals with sufficient eye control. Performance is much faster than most BCIs, but is very slow when compared to the speed of a good typist (80 words/min.).

The system architecture is advanced. The BRI is implemented on a separate processor with a Motorola 68000 CPU. A schematic of the system is shown in Figure 2.2. The BRI processor interacts with a special display showing the BRI grid of symbols as well as a speech synthesizer and special keyboard interface. The special keyboard interface enables the subject to control any regular PC programs that may be controlled from the keyboard. In addition, a remote control is interfaced with the BRI in order to enable the subject to control a TV or VCR. Since the BRI processor loads up all necessary software from the hard drive of a connected PC, the user may create or change command sequences. The main drawback of

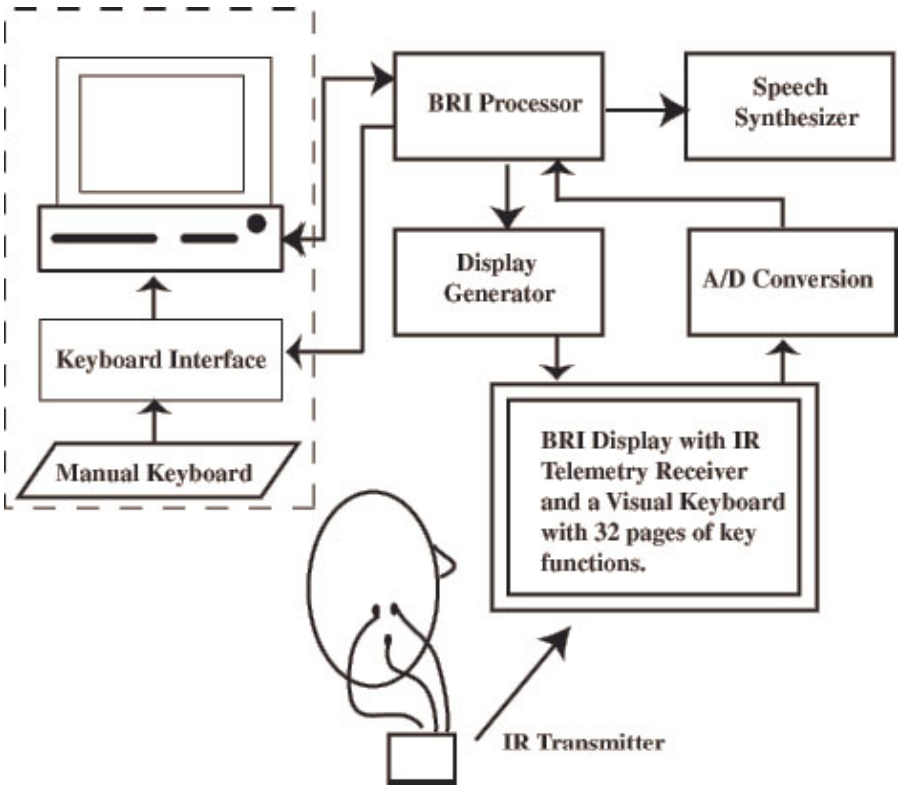


Figure 2.2: A schematic of the Brain Response Interface (BRI) system as described by Sutter.

the system architecture is that it is based on a special hardware interface. This may be problematic when changes need to be made to the system over time.

2.3.2 P3 Character Recognition

In a related approach, Farwell and Donchin use the P3 evoked potential [FD88]. A 6x6 grid containing letters from the alphabet is displayed on the computer monitor and users are asked to select the letters in a word by counting the number of times that a row or column containing the letter flashes. Flashes occur at about 10 Hz and the desired letter flashes twice in every set of twelve flashes. The average

response to each row and column is computed and the P3 amplitude is measured. Response amplitude is reliably larger for the row and column containing the desired letter. After two training sessions, users are able to communicate at a rate of 2.3 characters/min, with accuracy rates of 95%. This system is currently only used in a research setting.

A positive aspect of using a longer latency component such as the P3 is that it enables differentiating between when the user is looking at the computer screen or looking someplace else (as the P3 only occurs in certain stimulus conditions). Unfortunately, this system is also agonizingly slow, because of the need to wait for the appropriate stimulus presentation and because the stimuli are averaged over trials. While the experimental setup accomplishes its main goal of showing that the P3 may be used for a BCI interface, the subjective experiences of a subject with this system have yet to be considered. The 10 Hz rate of flashing may fatigue users as Sutter mentions and this rate of flashing may cause epilepsy in some subjects.

2.3.3 ERS/ERD Cursor Control

Pfurtscheller and his colleagues take a different approach [Nun95, PFK93, PKN+96, PNFP97, KFN+96]. Using multiple electrodes placed over sensorimotor cortex they monitor event-related synchronization/desynchronization (ERS/ERD) [PG99]. In all sessions, epochs with eye and muscle artifact are automatically rejected. This rejection can slow subject performance speeds.

As this is a research system, the user application is a simple screen that allows control of a cursor in either the left or right direction. In one experiment, for a single trial the screen first appears blank, then a target box is shown on one side

of the screen. A cross hair appears to let the user know that he/she must begin trying to move the cursor towards the box. Feedback may be delayed or immediate and different experiments have slightly different displays and protocols. After two training sessions, three out of five student subjects were able to move a cursor right or left with accuracy rates from 89-100%. Unfortunately, the other two students performed at 60% and 51%. When a third category was added for classification, performance dropped to a low of 60% in the best case [KFN+96].

The architecture of this BCI now contains a remote control interface that allows controlling the system over a phone line, LAN, or Internet connection. This allows maintenance to be done from remote locations. The system may be run from a regular PC, a notebook, or an embedded computer and is being tested for opening and closing a hand-orthesis in a patient with a C5 lesion. From this information, it appears that the user application must be independent from the BCI, although it is possible that two different BCI programs were constructed.

This BCI system was designed with the following requirements in mind [GSWP99]:

1. The system must be able to record, analyze, and classify EEG-data in real-time.

2. The classification results must have the ability to be used to control a device online.

3. The system must have the ability to have different experimental paradigms and give multimodal stimulations.

4. The system must display the EEG channels on-line on a monitor.

5. The system must store all data for later off-line analysis.

The system has the ability to record up to 96 channels of EEG simultaneously through the use of multiple A/D boards. Simulink and Matlab are the two software packages used: Simulink to calculate the parameters of the EEG state in real-time and Matlab to handle the data acquisition, timing, and experimental presentation. This design has the benefit of separating data processing from acquisition and application concerns. This may lead to greater encapsulation of data and maintainability. This design has the drawback of trying to use Matlab for both data acquisition and the BCI application. For simple applications such as the cursor control task, this decision makes sense. When the application becomes more complex this design decision may lead to problems. Matlab is not an object-oriented language and data encapsulation is not necessarily easy to accomplish. This may lead to poor maintainability. In addition, the system depends on Matlab for all program capabilities. This is fine for simple graphical interfaces, but may break down when the programmer wants to communicate with another program or even over the web. For these cases Matlab may offer several special program extensions, but buying many extensions becomes problematic and expensive. It would be easier to enable the application creator to use a variety of languages for the application.

2.3.4 A Steady State Visual Evoked Potential BCI

Middendorf and colleagues use operant conditioning methods in order to train volunteers to control the amplitude of the steady-state visual evoked potential (SSVEP) to florescent tubes flashing at 13.25 Hz [VWD96, MMCJ00, JMCM98]. This method of control may be considered as continuous as the amplitude may

change in a continuous fashion. Either a horizontal light bar or audio feedback is provided when electrodes located over the occipital cortex measure changes in signal amplitude. If the VEP amplitude is below or above a specified threshold for a specific time period, discrete control outputs are generated. After around 6 hours of training, users may have an accuracy rate of greater than 80% in commanding a flight simulator to roll left of right.

In the flight simulator, the stimulus lamps are located adjacent to the display behind a translucent diffusion panel. As operators increase their SSVER amplitude above one threshold, the simulator rolls to the right. Rolling to the left is caused by a decrease in the amplitude. A functional electrical stimulator (FES), has been integrated for use with this BCI. Holding the SSVER above a specified threshold for one second, causes the FES to turn on. The activated FES then starts to activate at the muscle contraction level and begins to increase the current, gradually recruiting additional muscle fibers to cause knee extension. Decreasing the SSVER for over a second, causes the system to deactivate, thus lowering the limb.

Recognizing that the SSVEP may also be used as a natural response, Middendorf and his colleagues have recently concentrated on experiments involving the natural SSVEP. When the SSVEP is used as a natural response, virtually no training is needed in order to use the system. The experimental task for testing this method of control has been to have subjects select virtual buttons on a computer screen. The luminance of the virtual buttons is modulated, each at a different frequency to produce the SSVEP. The subject selects the button by simply looking at it as in Sutters Brain Response Interface. From the 8 subjects participating in th experiment, the average percent correct was 92% with an average selection time

of 2.1 seconds. Middendorfs group has advocated using visual evoked potentials, in thi manner as opposed to their previous work on training control of the SSVEP, for multiple reasons. Using an inherent response means that less time is spent on training. The main drawback of this groups approach appears to be that they flicker ligh different frequencies. Sutter solved the problem of flicker-related fatigue by using alternating red/green illumination. The main frequency of stimulus presentation at 13.25 Hz may also cause epilepsy.

2.3.5 Mu Rhythm Cursor Control

Wolpaw and his colleagues free their subjects from being tied to a flashing florescent tube by training subjects to modify their mu rhythm [MNRW93, WMNF91]. This method of control is continuous as the mu rhythm may be altered in a continuous manner. It can be attenuated by movement and tactile stimulation as well as by imagined movement. A subject's main task is to move a cursor up or down on a computer screen. While not all subjects are able to learn this type of biofeedback control, the subjects that do perform with accuracy greater than or equal to 90%. These experiments have also been extended to two-dimensional cursor movement, but the accuracy of this is reported as having not reached this level of accuracy when compared to the one dimension control [VWD96].

Since the mu rhythm is not tied to an external stimulus, it frees the user from dependence on external events for control. The BCI system consists of a 64-channel EEG amplifier, two 32-channel A/D converter boards, a TMS320C30-based DSP board, and a PC with two monitors. One monitor is used by the subject and one by the operator of the system [MW03]. Only a subset of the 64-channels are

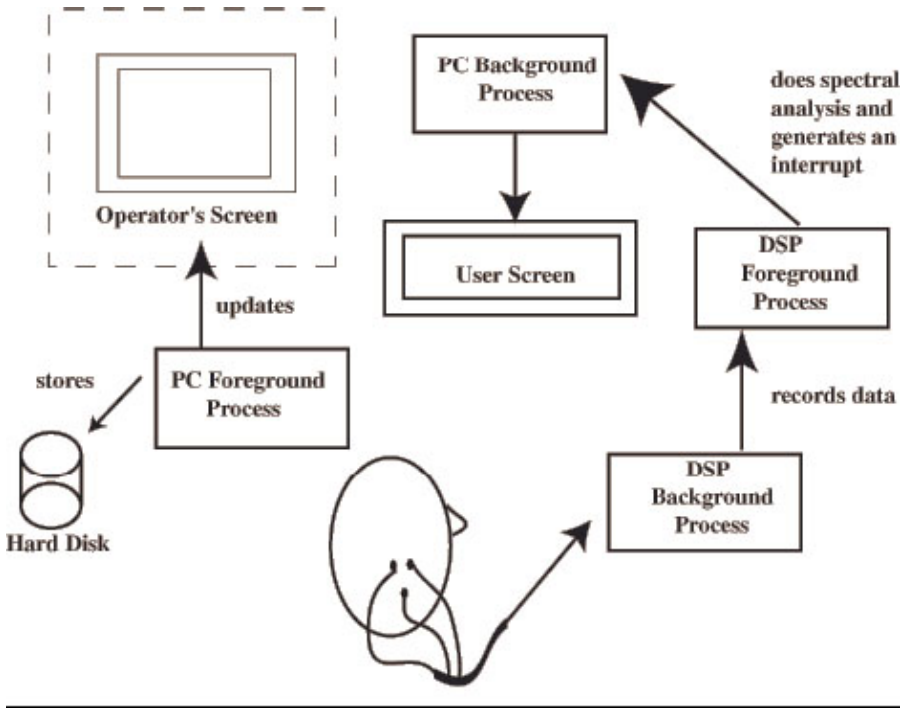


Figure 2.3: A schematic of the mu rhythm cursor control system architecture. The system contains four parallel processes. The PC foreground process must be linked to several of the other processes in order to obtain data, but these links have not been shown as they were not explicitly stated in the reference paper.

used for control, but the number of channels allows recognition to be adjusted to the unique topographical features of each subjects head. The DSP board is programmable in the C-language enabling testing of all program code prior to running it on the DSP board. Software is also programmed in C in order to create consistency across system modules. The architecture of the system is shown in Figure 2.3.

Four processes run between the PC and the DSP board. As signal acquisition occurs, an interrupt request is sent from the A/D board to the DSP at the end

of A/D conversion. The DSP then acquires the data from all requested channels sequentially and combines them to derive the one or more EEG channels that control cursor movement. This is the data collection process. A second process then takes care of performing a spectral analysis on the data. When this analysis is completed, the results are moved to dual-ported memory and an interrupt to the PC is generated. A background process on the PC then acquires spectral data from the DSP board and computes cursor movement information as well as records relevant trial information. This process runs at a fixed interval of 125 msec. The fourth process handles the graphical user interfaces for both the operator and the subject and records data to disk.

The separation of data collection and analysis enables different algorithms to be inserted for processing the EEG signals. All algorithms are written in C, which is much easier to program in than Assembly language, but is not as easy as the commercial Matlab scripting language and environment, which contains many helpful functions for mathematically processing data. The third and fourth processes contain design decisions that may make maintenance and flexibility difficult. The graphical user interface is tied to data storage. Conversion of EEG signals to cursor control numbers happens over the DSP foreground/background processes and in the PC background process. This lack of encapsulation promises to make changing the application and signal processing difficult if such changes are planned.

2.3.6 The Thought Translation Device

As another application used with severely handicapped individuals, the Thought Translation Device has the distinction of being the first BCI to enable an individual without any form of motor control to communicate with the outside world [BGH+99] . Out of six patients with ALS, three were able to use the Thought Translation Device. Of the other three, one lost motivation and later died and another discontinued use of the Thought Translation Device part way through training, and then later was unable to regain control. The paper implies that users do not want to use the BCI unless they absolutely must, but does not disambiguate subjective user satisfaction of the system from general user depression.

The training program may use either auditory or visual feedback. The slow cortical potential (see Table 2.1) is extracted from the regular EEG on-line, filtered, corrected for eye movement artifacts, and fed back to the patient. In the case of auditory feedback, the positivity/negativity of a slow cortical potential is represented by pitch. When using visual feedback, the target positivity/negativity is represented by a high and low box on the screen. A ball-shaped light moves toward or away from the target box depending on a subjects performance. The subject is reinforced for good performance with the appearance of a happy face or a melodic sound sequence.

When a subject performs at least 75% correct, he/she is switched to the language support program. At level one, the alphabet is split into two halves (letter-banks) which are presented successively at the bottom of the screen for several seconds. If the subject selects the letter-bank being shown by generating a slow

cortical potential shift, that side of the alphabet is split into two halves and so on, until a single letter is chosen. A return function allows the patient to erase the last written letter. These patients may now write email in order to communicate with other ALS patients world-wide. An Internet version of the thought translation device is under construction. The authors comment that patients refuse to use pre-selected word sequences because they feel less free in presenting their own intentions and thoughts.

2.3.7 An Implanted BCI

The implanted brain-computer interface system devised by Kennedy and colleagues has been implanted into two patients [KB98, KBM+00]. These patients are trained to control a cursor with their implant and the velocity of the cursor is determined by the rate of neural firing. The neural waveshapes are converted to pulses and three pulses are an input to the computer mouse. The first and second pulses control X and Y position of the cursor and a third pulse as a mouse click or enter signal.

The patients are trained using software that contains a row of icons representing common phrases (Talk Assist developed at Georgia Tech), or a standard 'qwerty' or alphabetical keyboard (Wivik software from Prentke Romich Co.). When using a keyboard, the selected letter appears on a Microsoft Wordpad screen. When the phrase or sentence is complete, it is output as speech using Wivox software from Prentke Romich Co. or printed text. There are two paradigms using the Talk Assist program and a third one using the visual keyboard. In the first paradigm, the cursor moves across the screen using one group of neural signals and down

the screen using another group of larger amplitude signals. Starting in the top left corner, the patient enters the leftmost icon. He remains over the icon for two seconds so that the speech synthesizer is activated and phrases are produced. In the second paradigm, the patient is expected to move the cursor across the screen from one icon to the other. The patient is encouraged to be as accurate as possible, and then to speed up the cursor movement while attempting to remain accurate. In the third paradigm, a visual keyboard is shown and the patient is encouraged to spell his name as accurately and quickly as possible and then to spell anything else he wishes.

This system uses commercially available software and thus the BCI implementation does not have to worry about maintenance of the user application. Unfortunately, the maximum communication rate with this BCI has been around 3 characters per minute. This is the same rate as quoted for EMG-based control with patient JR and is comparable with the rates achieved by externally-based BCI systems. Kennedy has founded Neural Signals, Inc. in order to help create hardware and software for locked-in individuals (see http://www.neuralsignals.com for more information) and the company is continually looking for methods to improve control. JR now has access to email and may be contacted through the email address shown on the companys web site.

Chapter 3

Kernel based hidden Markov model

In this chapter we address the problem of temporal signal classification. To enhance the performance of hidden Markov model (HMM), we present a dynamic model referred to as kernel based hidden Markov model (KHMM). The prominent feature which distinguishes our learning algorithm from traditional maximum likelihood based learning is that we develop a nonlinear discriminative procedure based on a maximum margin criterion to learn the model. As a nonparametric learning algorithm, our method has no need of prior knowledge of signal distribution while providing a strong generalization mechanism.

3.1 Introduction

Consider the problem of temporal signals classification, such as, EEG signals. Figure 3.1 shows an example the time course of the actual average EEG signal waveforms. The P300 potential is created in the central sites of EEG measurements when an infrequent and anticipated event occurs. The P300 signal is the signature

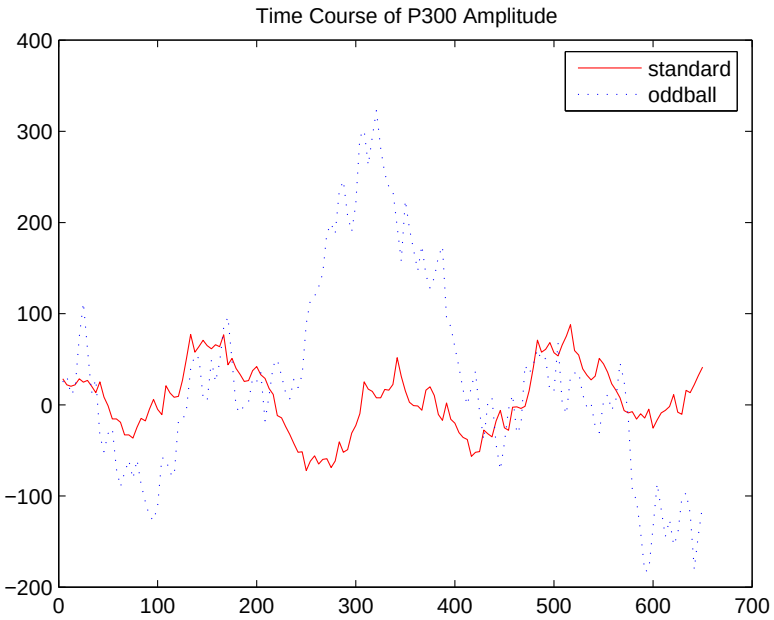


Figure 3.1: This figure shows the time course of the actual average signal waveforms (at Cz) or did not contain the desired character (standard)).

of the users brain registering the event, and typically occurs around 300 ms after the infrequent event takes place. A "'good"' P300 detector would reliably distinguish this impulse around $300ms$ with the other brain states. Previous study shows that the dynamic model of temporal signals and the concept of state transition can help us understand the signal well and develop the less error-prone classification algorithm [Rab89, OGNP01].

The hidden Markov model (HMM) is a statistical model that has been widely applied to many scientific and engineering areas [Rab89, OGNP01]. It can well model temporal or sequential structures of signals by combining the observation and state in an elegant manner. The most popular learning method for hidden Markov model is maximum likelihood (ML) estimation. The goal of ML estimation

is to find the parameter set that maximizes the likelihood of the training samples given their corresponding categories. By using the Baum-Welch algorithm, these parameters can be effectively estimated. However, ML estimation may not lead to an optimal performance. This is due in part to the mismatch between the chosen distribution form and the actual signal distribution that is typically not available.

To address this issue, a few recent endeavors resort to discriminative training approaches, such as maximum mutual information (MMI) estimation [BYB04] and minimum classification error (MCE) estimation [JCL97]. These approaches have their roots in maximum a posteriori (MAP) decision theory. Different from ML, here the learning is applied to all categories in the training phase. In the case of inadequate sparse training samples, they can usually demonstrate significant performance over the traditional ML approach. However, the performance of these learning methods still largely depends on consistency to actual data distribution.

We expect a nonparametric method that can be used with arbitrary distributions and without the assumption that forms of the underlying densities are known. Support vector machine (SVM), for example, is a nonparametric classification method with solid background in statistical learning theory [Vap98]. In principle, SVM constructs a hyperplane in the kernel space so as to maximize the margin of separation between positive and negative examples, which guarantees strong generalization compared with the traditional discriminative approaches used to train HMM models. However, SVM suffers from an apparent lack of considering the underlying process of signal generation so that it may fail to classify temporal signals.

Motivated by this dilemma, we proposed a dynamic discriminative model, referred to as kernel based hidden Markov model (KHMM). It incorporates kernel-based discriminative learning approaches into hidden Markov model, having no need of prior knowledge of signal distribution. In this chapter, we further propose a maximum margin discriminative learning method for KHMM. The learning is formulated as finding the maximum margin of separation between the category of the sample and the best runner-up in the kernel space. The formulation is by imposing the explicit constraint to the cost function so that the inferred state sequence from the designed model is the most possible state sequence.

3.2 Probabilistic models for temporal signal classification

Multiclass classification is to learn a function $h : \mathcal{X} \mapsto \mathcal{Y}$ that maps an instance x of $\mathcal{X}$ into an element y of $\mathcal{Y}$. In general $\mathcal{Y}$ is a countable set and has $\mathcal{Y} = \{1, \cdots, K\}$. In this thesis, we consider the problem of the signal classification where a signal $\mathbf{x}$ is a sequence from the set $\mathcal{X} = \{\mathcal{X}_1 \times \cdots \times \mathcal{X}_T\}$. In a motor imagery signal classification task [PN01], for example, the goal is to determine from the EEG signal, a time sequence signal for several seconds, which action the user is imagining.

3.2.1 Generative vs. Conditional

In general, probabilistic models can be subdivided into generative and conditional with respect to the prediction or classification task. A generative model assigns a

normalized joint density $p(\mathbf{x}, y)$ to the input and output space $\mathcal{X} \times \mathcal{Y}$ with

$$p(\mathbf{x}, y) \geq 0, \sum_{y \in \mathcal{Y}} \int_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}, y) = 1.$$

Correspondingly, a conditional model assigns a normalized density $p(y|\mathbf{x})$ only over the output space $\mathcal{Y}$ with

$$p(y|\mathbf{x}) \geq 0, \sum_{y \in \mathcal{Y}} p(y|\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \mathcal{X}.$$

Probabilistic interpretation of the model offers well-understood semantics and an immense toolbox of methods for inference and learning. It also provides an intuitive measure of confidence in the predictions of a model in terms of conditional probabilities. In addition, generative models are typically structured to allow very efficient maximum likelihood learning. A very common class of generative models is the exponential family:

$$p(\mathbf{x}, y) \propto \exp\{\mathbf{w}^T \mathbf{f}(\mathbf{x}, y)\}.$$

For exponential families, the maximum likelihood parameters $\mathbf{w}$ with respect to the joint distribution can be computed in closed form using the empirical basis function expectations $\mathbf{E}_S[\mathbf{f}(\mathbf{x}, y)$ [DeG70, HTF01].

Of course, this efficiency comes at a price. Any model is an approximation to the true distribution underlying the data. A generative model must make simplifying assumptions (more precisely, independence assumptions) about the entire $p(\mathbf{x}, y)$, while a conditional model makes many fewer assumption by focusing on $p(y|\mathbf{x})$. Because of this, by optimizing the model to fit the joint distribution $p(\mathbf{x}, y)$, we may be tuning the approximation away from optimal conditional distribution $p(y|\mathbf{x})$, which we use to make the predictions. Given sufficient data, the conditional model

will learn the best approximation to $p(y|\mathbf{x})$ possible using $\mathbf{w}$, while the generative model $p(\mathbf{x}, y)$ will not necessarily do so. Typically, however, generative models actually need fewer samples to converge to a good estimate of the joint distribution than conditional models need to accurately represent the conditional distribution. In a regime with very few training samples (relative to the number of parameters $\mathbf{w}$), generative models may actually outperform conditional models [NJ01].

3.2.2 Normalized vs. Unnormalized

Probabilistic semantics are certainly not necessary for a good predictive model if we are simply interested in the optimal prediction. Support vector machines, which do not represent a conditional distribution, typically perform as well or better than logistic regression [Vap98, CST00].

In general, we can often achieve higher accuracy models when we do not learn a normalized distribution over the outputs, but concentrate on the margin or decision boundary, the difference between the optimal y and the rest. Even more importantly, in many cases we discuss below, normalizing the model (summing over the entire $\mathcal{Y}$) is intractable, while the optimal y can be found in polynomial time. This fact makes standard maximum likelihood estimation infeasible. The learning methods we advocate in this thesis circumvent this problem by requiring only the maximization problem to be tractable. We still heavily rely on the representation and inference tools familiar from probabilistic models for the construction of and prediction in unnormalized models, but largely dispense with the probabilistic interpretation when needed.

3.3 Markov random field representation of dynamic model

A popular family of classification function h for the problem of the signal classification is statistically based. To achieve the minimum classification error, the optimal classifier, according to the classical Bayes decision theory, is the one that employs the decision rule of Eq. (3.1), which is called the *maximum a posteriori* (MAP) decision.

$$h(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} P(y|\mathbf{x}). \tag{3.1}$$

The decision rule, unfortunately, is not given in practice and has to be estimated from a training set with known class labels. A typical framework used to estimate these probabilities is the Hidden Markov model (HMM) [Rab89]. As a probabilistic graphical model, it is able to reveal the underlying process of signal generation even though the properties of the signal source (state) remain greatly unknown. As such, the a posterior probability is computed by summing over all possible state sequences $\mathbf{q} = [q_1, \cdots, q_T]^T$, that is

$$P(y|\mathbf{x}) = \sum_{\mathbf{q}} P(y, \mathbf{q}|\mathbf{x}). \tag{3.2}$$

Unfortunately, the calculation of a posterior probability requires to enumerate every possible state sequence of length T (the length of the signal). It is exponentially large and thus computationally unfeasible. In practice the correct state sequence, however, has the very high probability as opposed to the other state

sequences. As an alternative to Eq. (3.2), the a posterior probability can be approximated by only considering the most likely state sequence, that is[1]

$$P(y|\mathbf{x}) \approx \ P(\hat{\mathbf{q}}_{\mathbf{x}}|\mathbf{x}) \tag{3.3}$$

where $\hat{\mathbf{q}}_y$ is the most likely state sequence belong to the given observation sequence $\mathbf{x}$.

Given the observed signals up to time T, the conditional probability distribution $P(\mathbf{q}|\mathbf{x})$ is modeled with a Markov Random Field (MRF) [LMP01]. The structure of a Markov Random Field is defined by an undirected graph $\mathcal{G} = \{S, G\}$, where the nodes are associated with variables $S = \{q_1, \cdots, q_T\}$. A *clique* [GG84] is a set of nodes $c \subseteq S$ that form a fully connected subgraph (every two nodes are connected by an edge). Note that each subclique of a clique is also a clique. We denote an assignment of variables in a clique c as $\mathbf{q}_c$, and the space of all assignments to the clique as $\mathcal{Q}_c$. According to the theorem of random fields [LMP01], we define a conditional distribution associated with potential $V_c(\mathbf{q}_c, \mathbf{x})$ as:

$$P(\mathbf{q}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left[\sum_{c\in\mathcal{C}} V_c(\mathbf{q}_c, \mathbf{x})\right],$$

where $\mathcal{C}$ is the set of cliques for a graph and $Z(\mathbf{x})$ is the partial function given by $Z(\mathbf{x}) = \sum_{\mathbf{q}\in\mathcal{Q}} \exp[\sum_{c\in\mathcal{C}} V(\mathbf{q}_c, \mathbf{x})]$.

For simplicity, only first order Markov chain is used in our work. We consider a left-right directed graph and each node is a singleton clique. In the chain network in Figure 3.2, the cliques are simply the nodes and the edges: $\mathcal{C}(\mathcal{G}) = \{\{q_1\}, \cdots, \{q_T\}, \{q_1, q_2\}, \cdots, \{q_{T-1}, q_T\}\}$. Intuitively, the node potentials

[1] in Eq. (3.3), $p(\mathbf{q}|\mathbf{x}) = p(y, \mathbf{q}|\mathbf{x})$ because we can certainly identify every state sequence $\mathbf{q}$ as the unique category.

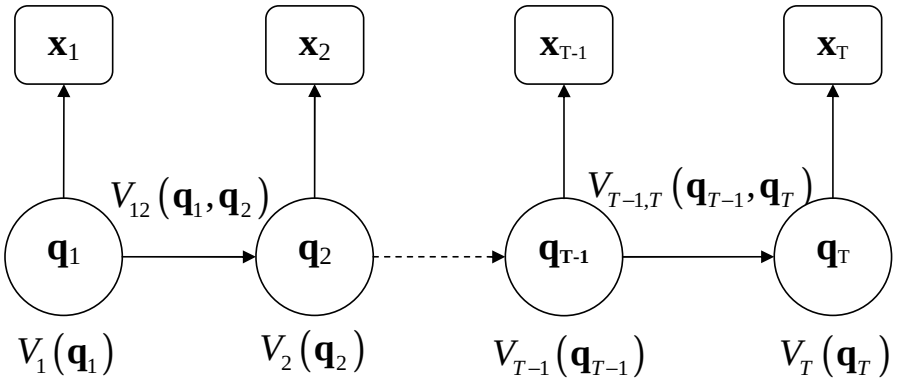


Figure 3.2: First order Markov chain

quantify the correlation between the input $\mathbf{x}$ and the value of the node, while the edge potentials quantify the correlation between the pair of adjacent state variables as well as the input $\mathbf{x}$. Potentials do not have a local probabilistic interpretation, but can be thought of as defining an unnormalized score for each assignment in the clique. Conditioned on the signal input, appropriate node potentials in our network should give high scores to the most possible hidden state chains. For simplicity, assume that the edge potentials would not depend on the input signals, but simply should give high scores to pairs of hidden states that tend to appear often consecutively.

In fact, a Markov random field is a generalized log-linear model, since the potentials $V_c(\mathbf{q}_c, \mathbf{x})$ could be represented as a sum of basis functions over $\mathbf{x}, \mathbf{q}_c$:

$$V_c(\mathbf{q}_c, \mathbf{x}) = \sum_{k=1}^{n_c} w_{c,k} f_{c,k}(\mathbf{q}_c, \mathbf{x}) = \mathbf{w}_c^T \mathbf{f}_c(\mathbf{q}_c, \mathbf{x})$$

where n_c is the number of basis functions for the clique c. Hence the log of the conditional probability is given by:

$$\log P(y|\mathbf{x}) \approx \sum_{c \in \mathcal{C}(\mathcal{G})} \mathbf{w}_c^T \mathbf{f}_c(\mathbf{q}_{\mathbf{x},c}, \mathbf{x}) - \log Z(\mathbf{x}). \tag{3.4}$$

In case of node potentials for temporal signal classification, we define the basis function as the kernel features of the observation $\mathbf{x}_t$, given the state q_t, we denote it as $\boldsymbol{\phi}(q_t, \mathbf{x}_t)$. For the edge potentials, we can define basis functions as the indicator functions where $f_{t-1,t}(q_{t-1}, q_t) = 1$ only if the transition from state q_{t-1} to state q_t is allowed. In this problem, as well as many others, we are likely to share the weights of the model $\mathbf{w}_c$ across cliques. Usually, all of node potentials would share the same weights and basis functions and similarly for the pairwise cliques, no matter in what position they appear in the graph.

We define a vector $\mathbf{f}$ to replace all the basis functions above, simplifying the decision rule. For the first-order Markov model, $\mathbf{f}$ has node functions and edge functions, so when the clique c is a node, the edge functions in $f(\mathbf{q}_c, \mathbf{x})$ are defined to evaluate to zero. Similarly, when the clique c is an edge, the node functions in $f(\mathbf{q}_c, \mathbf{x})$ are also defined to evaluate to zero. Now we can write:

$$\mathbf{f}(\mathbf{q}, \mathbf{x}) = \sum_{c \in \mathcal{C}(\mathcal{G}|_y)} \mathbf{f}(\mathbf{q}_c, \mathbf{x}).$$

The weights $\mathbf{w}$ can be defined in the corresponding manner, so the classifier according to our proposed model is given by:

$$h(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \mathbf{w}^T \mathbf{f}(\mathbf{q}_\mathbf{x}, \mathbf{x}). \tag{3.5}$$

3.4 Inference

There are several important questions that can be answered by probabilistic models. The task of finding the most likely assignment, known as maximum a-posteriori (MAP) or most likely explanation (MPE), is just one of such questions, but most relevant to our discussion. This problem is solved straightforward in the case of

the normal pattern classification tasks, since there is no more unknown variable once the model is built via the training samples. The temporal signal classification, however, has to attempt to recover all the states associated with the given observation sequence before finding assignment.

It should be clear that there is no "'correct"' states sequence to be found. Hence for practical situations, an optimality criterion is usually used to solve this problem as best as possible. The difficulty lies with the definition of the optimal state sequence. That is, there are several possible optimality criteria. For example, one possible optimality criterion is to choose the state q_t that are individually most likely at each time t. This optimality criterion maximizes the confidence of correct individual states. To implement this solution, we can define the a posteriori probability variable with logarithm form

$$\gamma_t(i) = \log P(q_t = i|\mathbf{x}), \tag{3.6}$$

that is, the logarithm of probability of being in state i at time t, given the observation sequence $\mathbf{x}$. According to Eq. (3.4), we can express γ_t as the following form:

$$\gamma_t(i) = \mathbf{w}^T\mathbf{f}(i, \mathbf{x}),$$

where $\mathbf{w}$ is the weight for the dynamic model. Using $\gamma_t(i)$, we can solve for the individually most likely state $\hat{q}_t$ at time t, as

$$\hat{q}_t = \arg\max_{1\leq i\leq M} \gamma_t(i), \qquad 1 \leq t \leq T. \tag{3.7}$$

Although Eq. (3.7) maximizes the note potential of correct states (by choosing the most likely state for each t), there could be some problems with the resulting state sequence. For example, when the dynamic model has state transitions which are

not allowed ($f_t(i, j, \mathbf{x}) = 0$ for some i and j), the "'optimal"' state sequence may, in fact, not even be a valid state sequence. This is because the solution of Eq (3.7) simply determines the most likely state at every instant, without regard to the edge potential associated with the note t.

One possible solution to the above problem is to modify the optimality criterion by involving the associated edge cliques. For example, one could solve for the state sequence that maximizes the confidence of correct pairs of states (q_{t-1}, q_t), or triples of states (q_{t-2}, q_{t-1}, q_t), etc. Although these might be applicable for some applications, the most widely used criterion, also used in our work, is to find the single best state sequence, that is

$$\hat{\mathbf{q}} = \arg\max_{q \in \mathcal{Q}_y} P(\mathbf{q}|\mathbf{x}).$$

This problem can be solved by the Viterbi dynamic programming in $\mathcal{O}(L)$ time [Vit67, For73]. Let the highest score of any subsequence from q_1 to q_{t-1} ending with state i be defined as

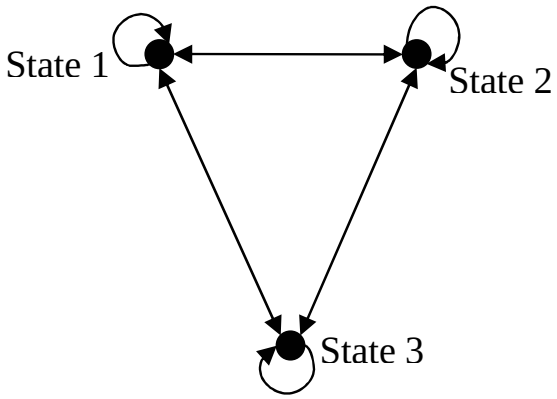
$$\delta_t(i) = \max_{q_1, \cdots, q_{t-1}} \left[\sum_{t'=1}^{t-1} \mathbf{w}^T \mathbf{f}(\mathbf{q}_{t'}, \mathbf{x}) + \sum_{t'=1}^{t-1} \mathbf{w}^T \mathbf{f}(q_{t'}, q_{t'+1}, \mathbf{x}) \right].$$

By induction we have

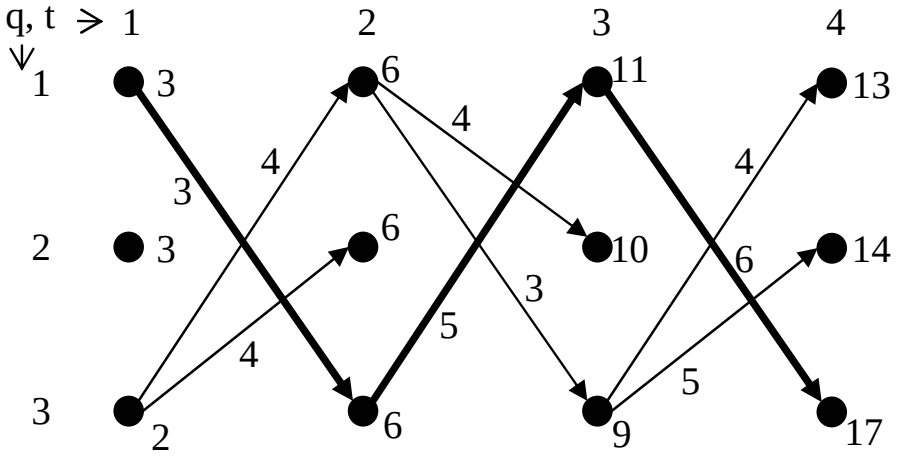
$$\delta_{t+1}(j) = \max_i \left[\delta_t(i) + \mathbf{w}^T \mathbf{f}(i, j, \mathbf{x}) + \mathbf{w}^T \mathbf{f}(j, \mathbf{x}) \right]. \tag{3.8}$$

To actually retrieve the state sequence, we need to keep track of the argument that maximized Eq (3.8), for each t and j. We do this via the array $\psi_t(j)$.

We shall illustrate with an example on how the maximization is done. Consider a three state KHMM. Figure 3.3a shows the possible paths of state transition as described above, while Figure 3.3b is the procedure to find the most likely path for the simple example.



(a)



(b)

Figure 3.3: Illustration of Viterbi searching: (a) The state machine. (b) How a path of maximum score is traced out using the Viterbi algorithm. The final maximum score path is shown in bold.

Consider Figure 3.3b. We take an observation sequence $\mathbf{x}_1, \cdots \mathbf{x}_4$ of length four. At time $t = 2$ state 1 can be visited from any of the three states that we had at time $t = 1$. We find out the potentials on each of these notes and add corresponding edge potentials to the initial confidences (we shall call this cumulative potentials at state 1 as the score at state 1 at time $t = 2$). Thus going from state 1 to state 1, the score at state 1 is $3 + 1 = 4$. Similarly the score at state 1 in going from state 2 to state 1 is $3 + 2 = 5$ and the score in going from state 3 to state 1 is $2 + 4 = 6$. Of these the score 6 is maximum. Hence we retain this score at state 1 for further calculations. The maximum cumulative score paths are shown in the figure by arrowed lines. The cumulative scores have been shown alongside the respective states at each time instant. We repeat the same procedure for state 2 and state 3. We see that the maximum scores at state 2 and state 3 are 6 (through state 3) and 6 (through state 1) respectively.

We repeat the above procedure again for $t = 3$ but now using the scores calculated above for each state rather than the initial confidences (we used them above because we started with $t = 1$). And the same procedure is repeated for $t = 4$. Now select out the state which has maximum score of all the states. We see that state 3 is the required state with a score of 17. Back tracing the sequence of states through which we got at state 3 at time $t = 4$ gives the required sequence of states through which the given observation sequence has highest confidence of occurrence. As can be seen from the figure this state sequence is state 1 ,state 3 ,state 1 ,state 3. This sequence has been shown in bold in the figure.

To prove our point suppose you were given that the length of observation sequence is required to be two and that the last state is to be state 3. What path

would you choose to maximize the cost? The procedure outlined above clearly shows that we would choose the path beginning at state 1 and ending at state 3 (at $t = 2$) as that gave the maximum score cost (viz. 6) at state 3. All other paths have a lower score. Similar argument applies if any other state were required to be the last state. Similarly if the observation sequence were to be of length three and ending in say state 1 we would choose the path state 1, state 3, state 1 as outlined in the figure and described above. This means that at any given instant the path tracked up to any state by the above procedure is the maximum score path if we were to stop at that instant at that state. Proceeding to $t = 4$ we see that we have the maximum score paths corresponding to stopping in state 1, state 2 or state 3 respectively. We just pick up the highest of these three because we are interested in the maximum score and hence we choose to stop in state 3 which gives us the maximum score. The complete procedure for finding the most likely state sequence is stated in Figure 3.4.

3.5 Maximum margin discriminative learning

Support vector machine, as one of the most important applications of statistical learning theory, is originally designed for the binary classification problem. It has a nice geometrical interpretation of discriminating one class from the other by a hyperplane with the maximum margin. Maximum Margin discriminative learning can be used to find this optimal decision surface, increasing the "confidence" of the classification. However, It is not straightforward that defining the margin in the case of multi-category classification problem. Currently, there are two types of approaches for multi-class SVM. One is by constructing and combining several

1. **Initialization**

$$\delta_1(i) = \mathbf{w}^T \mathbf{f}(i, \mathbf{x})$$

$$\psi_1(i) = 0, \qquad 1 \leq i \leq M$$

2. **Recursion**

$$\delta_t(j) = \max_{1 \leq i \leq M} \left[\delta_{t-1}(i) + \mathbf{w}^T \mathbf{f}(i, j, \mathbf{x})\right] + \mathbf{w}^T \mathbf{f}(j, \mathbf{x})$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq M} \left[\delta_{t-1}(i) + \mathbf{w}^T \mathbf{f}(i, j, \mathbf{x})\right], \qquad 2 \leq t \leq T, 1 \leq j \leq M$$

3. **Termination**

$$\log P(\hat{\mathbf{q}}) = \max_{1 \leq i \leq M} \delta_T(i)$$

$$\hat{q}_T = \arg \max_{1 \leq i \leq M} \delta_T(i)$$

4. **Backtracking**

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \qquad t = T-1, \cdots, 1$$

Figure 3.4: The complete inference algorithm

binary classifiers [BCD+94, Kre99, PCSt00], while other is by directly considering all data in one optimization formulation [WW99, CS01]. We define our margin in a similar way as Crammer and Singer [CS01], to construct the decision surface in such a way that the margin between the true class and the best runner-up is maximized. Therefore, given the training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the margin r has the upper bound as follows:

$$r \leq \min_{i} \left\{ \max_{\mathbf{q} \in \mathcal{Q}|_{y_i}} [\mathbf{w}_{y_i} \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i)] - \max_{k \neq y_i} \left[\max_{\mathbf{q} \in \mathcal{Q}|_k} [\mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i)] \right] \right\} \tag{3.9}$$

where $\mathbf{f}(\mathbf{q}, \mathbf{x})$ is the basis function defined in previous section.

Unfortunately, it may be difficult to maximize the margin of separation directly. Similar to the support vector machine, this optimization problem is equivalent to minimizing the Euclidean norm of the weight vector $\mathbf{w}$ while keeping the margin $r = 1$. In consequence, the conditional probability of the most possible state sequence for the correct class, given the optimal weights, is larger by at least one than the probabilities assigned to the rest of the state sequences. Mathematically,

$$\forall i, k, \mathbf{q} \qquad \mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) + (1 - \delta_{k,y_i}) \leq \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) \tag{3.10}$$

where $\hat{\mathbf{q}}_i = \arg\max_{\mathbf{q}} [\mathbf{w}_{y_i} \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i)]$ is the most possible state for the correct class.

In the case of violating the condition defined by Eq. (3.10), we have to suffer a loss which is linearly proportional to the difference between the confidence of the correct label and the maximum among the confidences of the other labels. A graphical illustration of the above is given in Figure 3.5. The circles in the figure denote different labels and the correct label is plotted in dark while the rest of the labels are plotted in blank. The height of each label designates its confidence. Three settings are plotted in the figure. The left plot (a) corresponds to the case

when the margin is not less than one, and therefore the condition is hold, and hence the example is correctly classified. The middle figure (b) shows a case where the example is correctly classified but with a small margin and we suffer some loss. The right plot depicts the loss of a misclassified example. In summary, our goal is to develop a computationally efficient procedure for using the training sample $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ to find the optimal hyperplane such that above loss is equal to zero for all the samples. When the sample $\mathcal{S}$ is linearly separable by a kernel based hidden Markov model, we seek all the matrices $\mathbf{w}_k{}_{k=1}^K$ of the smallest summation of norm that satisfies Eq. (3.10). The result is the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\sum_k \|\mathbf{w}_k\|_2^2 \\ \text{subject to} : \forall i, k, \mathbf{q} \quad & \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) + \delta_{k,y_i} - \mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \geq 1. \end{aligned} \tag{3.11}$$

Note that $N \times M^T$ of the constraints for $k = y_i$ are automatically satisfied since,

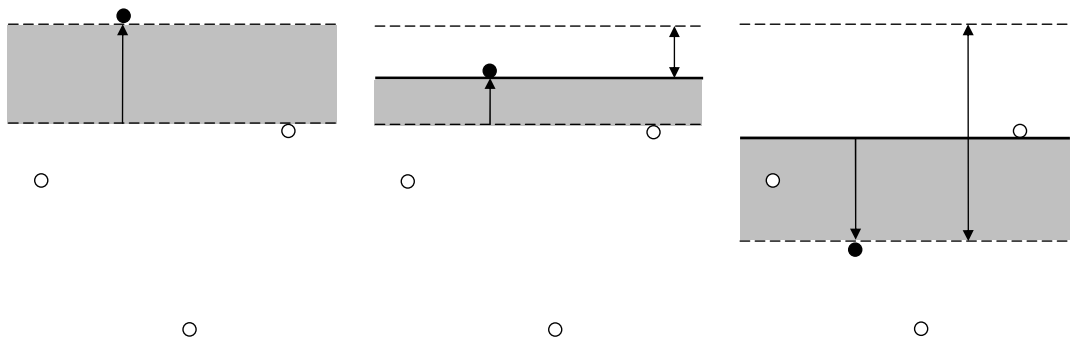
$$\forall \mathbf{q} \quad \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) + \delta_{y_i,y_i} - \mathbf{w}_{y_i} \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \geq 1,$$

where M is the number of states and T the length of temporal signal.

This property is an artifact of the linearly separable case. In the case of non-separable patterns, however, it is not possible to construct a separating hyperplane without encountering classification errors. In order to allow some constraints to be violated, we introduce a new set of nonnegative slack variables $\{\xi_i\}_{i=1}^N$. Therefore, the constrained optimization problem that we have to solve may now be stated as:

Given the training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$*, find the optimum values of the weight vector* $\mathbf{w}$ *such that they satisfy the constraints*

$$\forall i, k, \mathbf{q} \qquad \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) + \delta_{k,y_i} - \mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i), \geq 1 - \xi_i$$



(a) correct classification (b) correct with loss (c) misclassification

Figure 3.5: Illustration of the margin bound employed by the optimization problem

and the weight vector $\mathbf{w}$ *minimizes the cost function:*

$$J(\mathbf{w}) = \frac{1}{2}\sum_{k} \|\mathbf{w_k}\|_2^2 + C\sum_{i} \xi_i,$$

Where $C > 0$ is a regularization constant and is determined experimentally.

To solve this optimization problem, the correct path $\hat{\mathbf{q}}$ has to be estimated in advance. We will discuss this obstacle in the next chapter.

3.6 Conclusion

In this chapter, we presented here a kernel based hidden Markov model for classifying multi-class temporal signal data. The model is capable of both exploring the temporal dynamic of the signals and maximizing the margins between classes in an efficient way, by taking advantage of the rich language of Markov model and the kernel techniques.

Because our approach only relies on using the maximum in the model for prediction, and does not require a normalized distribution $P(y|\mathbf{x})$ over all outputs,

maximum margin estimation can be tractable when maximum likelihood is not. It can be formulated as a compact QP with linear constraints, which we will explore in the next chapter. An additional advantage of our approach is that the solutions to the estimation are relatively sparse in the dual space, which makes the use of kernels much more efficient.

Chapter 4

KHMM algorithms and experiments

In the previous chapter we introduce the kernel based hidden Markov model for temporal signal classification. The chief computational bottleneck in learning the parameters of model is due to a complete absence of the labels of states. To address this problem, this chapter present a two-step learning algorithm that alternatively estimates the parameters of the designed model and the most possible state sequences until convergence. The convergence of this algorithm has been proved in this paper. Furthermore, we provide a set of the compact formulations equivalent to the dual problem of our proposed framework, which dramatically reduces the exponentially large optimization problem to polynomial size.

We apply our KHMM framework and two-step learning algorithm to a set of synthetic data sequences with mixture of Gaussian. We show that our models significantly outperform the traditional HMM approach by incorporating high-dimensional decision boundaries of RBF kernels while capturing dynamic structure of temporal signals.

4.1 Two-step learning algorithm

Because the underlying stochastic process is not usually observable and thus the optimal state sequence has to be estimated, the constrained optimization problem given in section 3 can not be solved directly using standard quadratic programming (QP) techniques. In this section, we present a two-step learning algorithm for solving the constrained optimization problem. It can be seen that this two-step algorithm is similar to the mathematics of standard Expectation-Maximization (EM) technique [DLR77], although our optimization problem is not directly related to probability estimation.

The EM algorithm is an iterative optimization technique to solve the parameters estimation problem while we are not given some "hidden" nuisance variables. In particular, an auxiliary function which averages over the values of the hidden variables given the parameters at the previous iteration is defined. By minimizing this auxiliary function, we will always carry out an improvement over the previous estimated parameters, unless finding the optimal values of parameters. In our case, the hidden variables are the most possible state sequences $\hat{\mathbf{q}}_i$. Instead of considering the expected values over the distribution on these unobservable state sequences, we just consider the sequences of states that minimize the cost, given the previous values of the parameters:

$$\begin{aligned} Q(\mathbf{w}, \bar{\mathbf{w}}) \stackrel{\texttt{def}}{=} & \frac{1}{2} \sum_k \|\mathbf{w}_k\|_2^2 + C \sum_i \xi_i \\ & + \sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \left[\mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) - \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\bar{\mathbf{w}}_{y_i}), \mathbf{x}_i) - \delta_{y_i,k} + 1 - \xi_i \right] \end{aligned} \tag{4.1}$$

where the auxiliary nonnegative variables $\eta_{i,k,\mathbf{q}}$ are Lagrange multipliers, and $\hat{\mathbf{q}}_i(\bar{\mathbf{w}}_{y_i})$ is the most possible state sequence of the sample $\mathbf{x}_i$ given the previous value of

weight $\bar{\mathbf{w}}_{y_i}$.

4.1.1 Derivation of reestimation formulas from the Q-function

The next step is to find a new set of weights $\mathbf{w}$ which minimizes $Q(\mathbf{w}, \bar{\mathbf{w}})$ where $\bar{\mathbf{w}}$ is the previous set of weights. To solve this optimization subproblem, we use Karush-Kuhn-Tucker (KKT) theorem [Ber95]. Accordingly, the solution to the optimization subproblem given in Eq. (4.1) is determined by the the saddle point of the function Q, which would be minimized with respect to $\mathbf{w}$ and ξ; it also would be maximized with respect to η. Thus, the minimum over the variables ξ requires the following condition of optimality:

$$\frac{\partial Q}{\partial \xi_i} = C - \sum_{k,\mathbf{q}} \eta_{i,k,\mathbf{q}} = 0. \tag{4.2}$$

Application of this optimality condition yields:

$$\sum_{k,\mathbf{q}} \eta_{i,k,\mathbf{q}} = C. \tag{4.3}$$

Combining with the constraint $\eta_{i,k,\mathbf{q}} \geq 0$, it leads to the following constraint:

$$0 \leq \eta_{i,k,\mathbf{q}} \leq C. \tag{4.4}$$

Similarly, for $\mathbf{w}$ we require,

$$\frac{\partial Q}{\partial \mathbf{w}_k} = \mathbf{w}_k + \sum_{i,\mathbf{q}} \eta_{i,k,\mathbf{q}} \mathbf{f}(\mathbf{q}, \mathbf{x}_i) - \sum_i \left[\delta_{y_i,k} \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) \sum_{k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \right] = 0. \tag{4.5}$$

Substituting Eq. (4.3) into Eq. (4.5) and representing $\mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) = \sum_{\mathbf{q}} \delta_{\hat{\mathbf{q}}_i,\mathbf{q}} \mathbf{f}(\mathbf{q}, \mathbf{x}_i)$, we have,

$$\mathbf{w}_k = \sum_{i,\mathbf{q}} \left(C \delta_{y_i,k} \delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \eta_{i,k,\mathbf{q}} \right) \mathbf{f}(\mathbf{q}, \mathbf{x}_i). \tag{4.6}$$

Let $\alpha_{i,k,\mathbf{q}} = C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \eta_{i,k,\mathbf{q}}$. Then Eq. (4.6) that describes the form of $\mathbf{w}$ becomes:

$$\mathbf{w}_k = \sum_{i,\mathbf{q}} \alpha_{i,k,\mathbf{q}} \mathbf{f}(\mathbf{q}, \mathbf{x}_i). \tag{4.7}$$

Additionally, we have the following constraint:

$$\begin{aligned} \sum_{k,\mathbf{q}} \alpha_{i,k,\mathbf{q}} &= \sum_{k,\mathbf{q}} (C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \eta_{i,k,\mathbf{q}}) \\ &= C - \sum_{k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \\ &= 0. \end{aligned} \tag{4.8}$$

To postulate the dual problem for our optimization subproblem, we first expand Eq. (4.1) as follows:

$$\begin{aligned} Q(\mathbf{w}, \bar{\mathbf{w}}) = \frac{1}{2}\sum_k \|\mathbf{w}_k\|_2^2 + \sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) - \sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) \\ + \sum_i \xi_i \left(C - \sum_{k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \right) + \sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \left(1 - \delta_{y_i,k}\right). \end{aligned} \tag{4.9}$$

The fourth term on the right-hand side of Eq. (4.9) is zero by virtue of the optimality condition of Eq. (4.3). Furthermore, from Eqs. (4.7) and (4.8), and the definition of α we rewrite other terms of Eq. (4.9) as follows:

$$\begin{aligned} \sum_k \|\mathbf{w}_k\|_2^2 &= \sum_k \left(\sum_{i,\mathbf{q}} \alpha_{i,k,\mathbf{q}} \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \right) \cdot \left(\sum_{j,\mathbf{q}'} \alpha_{j,k,\mathbf{q}'} \mathbf{f}(\mathbf{q}', \mathbf{x}_j) \right) \\ &= \sum_k \sum_{i,j,\mathbf{q},\mathbf{q}'} \alpha_{i,k,\mathbf{q}} \alpha_{j,k,\mathbf{q}'} K(\mathbf{q}, \mathbf{x}_i, \mathbf{q}', \mathbf{x}_j), \end{aligned}$$

$$\begin{aligned} &\sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \\ =&\sum_{i,k,\mathbf{q}} \left\{ (C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \alpha_{i,k,\mathbf{q}}) \left[\sum_{j,\mathbf{q}'} \alpha_{j,k,\mathbf{q}'} \mathbf{f}(\mathbf{q}', \mathbf{x}_j) \right] \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \right\} \\ =&C \sum_{i,j,\mathbf{q}} \alpha_{j,y_i,\mathbf{q}} K(\hat{\mathbf{q}}_i, \mathbf{x}_i, \mathbf{q}, \mathbf{x}_j) - \sum_k \sum_{i,j,\mathbf{q},\mathbf{q}'} \alpha_{i,k,\mathbf{q}} \alpha_{j,k,\mathbf{q}'} K(\mathbf{q}, \mathbf{x}_i, \mathbf{q}', \mathbf{x}_j), \end{aligned}$$

$$
\begin{aligned}
&\sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} \mathbf{w}_{y_i} \cdot \mathbf{f}(\bar{\mathbf{q}}_i, \mathbf{x}_i) \\
=&\sum_{i,k,\mathbf{q}} \left\{ (C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \alpha_{i,k,\mathbf{q}}) \left[\sum_{j,\mathbf{q}'} \alpha_{j,y_i,\mathbf{q}'} \mathbf{f}(\mathbf{q}', \mathbf{x}_j) \right] \cdot \mathbf{f}(\hat{\mathbf{q}}_i, \mathbf{x}_i) \right\} \\
=&C \sum_{i,j,\mathbf{q}} \alpha_{j,y_i,\mathbf{q}} K(\hat{\mathbf{q}}_i, \mathbf{x}_i, \mathbf{q}, \mathbf{x}_j) - \sum_{i,j,\mathbf{q}'} \left[\alpha_{j,y_i,\mathbf{q}'} K(\hat{\mathbf{q}}_i, \mathbf{x}_i, \mathbf{q}', \mathbf{x}_j) \left(\sum_{k,\mathbf{q}} \alpha_{i,k,\mathbf{q}} \right) \right] \\
=&C \sum_{i,j,\mathbf{q}} \alpha_{j,y_i,\mathbf{q}} K(\hat{\mathbf{q}}_i, \mathbf{x}_i, \mathbf{q}, \mathbf{x}_j), \\
&\sum_{i,k,\mathbf{q}} \eta_{i,k,\mathbf{q}} (1 - \delta_{y_i,k}) = \sum_{i,k,\mathbf{q}} (C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}} - \alpha_{i,k,\mathbf{q}}) (1 - \delta_{y_i,k}) = \sum_{i,k,\mathbf{q}} \delta_{y_i,k} \alpha_{i,k,\mathbf{q}},
\end{aligned}
$$

where we define $K(\cdot) = \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \cdot \mathbf{f}(\mathbf{q}', \mathbf{x}_j)$.

Therefore, we obtain the following optimization subproblem in the dual formulation as follows:

$$
\begin{aligned}
&\max_{\alpha} \left\{ -\frac{1}{2} \sum_{k} \sum_{i,\mathbf{q}} \sum_{j,\mathbf{q}'} \alpha_{i,k,\mathbf{q}} \alpha_{j,k,\mathbf{q}'} K(\mathbf{q}, \mathbf{x}_i, \mathbf{q}', \mathbf{x}_j) + \sum_{i,k,\mathbf{q}} \alpha_{i,k,\mathbf{q}} \delta_{y_i,k} \right\} \\
&s.t. \quad \sum_{k,\mathbf{q}} \alpha_{i,k,\mathbf{q}} = 0, \forall i; \quad \alpha_{i,k,\mathbf{q}} \leq C\delta_{y_i,k}\delta_{\hat{\mathbf{q}}_i,\mathbf{q}}, \forall i, k, \mathbf{q};
\end{aligned}
\tag{4.10}
$$

Having determined the optimum Lagrange multipliers, denoted by $\alpha_{i,k,\mathbf{q}}$, we may compute the optimum weights $\mathbf{w}$, yielding:

$$
\mathbf{w}_k = \sum_{i} \sum_{\mathbf{q} \in \mathcal{Q}|_k} \alpha_{i,k,\mathbf{q}} \mathbf{f}(\mathbf{q}, \mathbf{x}_i) \tag{4.11}
$$

where $\mathcal{Q}|_k$ is the subset of state sequences $\mathbf{q}$ which belongs to model k.

4.1.2 Convergence

The algorithm consists of steps of repeatedly replacing $\bar{\mathbf{w}}$ by $\mathbf{w}$ using update Eq. (4.11) until convergence. Theorem 4.2 guarantees that such an approach will converge in a finite number of iterations to a solution so that the cost function

$J(\mathbf{w})$ reaches the minimal point. Note that the algorithm leads to local minima only. To give the proof of theorem 4.2, we first prove the following lemma,

Lemma 4.1. *For any pair* $(\mathbf{w}, \bar{\mathbf{w}})$ *in* $\Omega \times \Omega$*,*

$$J(\mathbf{w}) - Q(\mathbf{w}, \bar{\mathbf{w}}) \leq 0,$$

with equality if and only if $\mathbf{w} = \bar{\mathbf{w}}$*.*

Proof. From the KKT condition of quadratic optimization [Ber95], the third term of $Q(\mathbf{w}, \bar{\mathbf{w}})$ must be equal to zero. Thus,

$$\begin{aligned} J(\mathbf{w}) - Q(\mathbf{w}, \bar{\mathbf{w}}) &= \left[\frac{1}{2}\sum_k \|\mathbf{w}_k\|_2^2 + C\sum_i \xi_i'\right] - \left[\frac{1}{2}\sum_k \|\mathbf{w}_k\|_2^2 + C\sum_i \xi_i\right] \\ &= C\sum_i (\xi_i' - \xi_i) \end{aligned}$$

where $\{\xi_i'\}$ and $\{\xi_i\}$ are the set of slack variables in the function $J(\mathbf{w})$ and $Q(\mathbf{w}, \bar{\mathbf{w}})$, respectively. According to the definition of slack variables, we can compute $\{\xi_i'\}$ and $\{\xi_i\}$ as follows:

$$\begin{aligned} \xi_i' &= \max_k \left\{\mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) - \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\mathbf{w}_{y_i}), \mathbf{x}_i) - \delta_{y_i,k} + 1\right\}. \\ \xi_i &= \max_k \left\{\mathbf{w}_k \cdot \mathbf{f}(\mathbf{q}, \mathbf{x}_i) - \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\bar{\mathbf{w}}_{y_i}), \mathbf{x}_i) - \delta_{y_i,k} + 1\right\}. \end{aligned}$$

Since $\hat{\mathbf{q}}$ is by definition the most possible state sequence, we have the inequality $\mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\mathbf{w}_{y_i}), \mathbf{x}_i) \geq \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\bar{\mathbf{w}}_{y_i}), \mathbf{x}_i)$. Thus,

$$\begin{aligned} J(\mathbf{w}) - Q(\mathbf{w}, \bar{\mathbf{w}}) &= \sum_i (\xi_i' - \xi_i) \\ &= \sum_i \left[\mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\bar{\mathbf{w}}_{y_i}), \mathbf{x}_i) - \mathbf{w}_{y_i} \cdot \mathbf{f}(\hat{\mathbf{q}}_i(\mathbf{w}_{y_i}), \mathbf{x}_i)\right] \leq 0. \end{aligned}$$

and the identity only holds when $\mathbf{w} = \bar{\mathbf{w}}$. □

Theorem 4.2. *Suppose that* $\mathbf{w}^{(p)}$ *for* $p = 0, 1, 2, \ldots$ *is an instance of the two-step learning algorithm such that:*

1. *the sequence* $J(\mathbf{w}^{(p)})$ *is bounded, and*

2. $Q(\mathbf{w}^{(p+1)}, \mathbf{w}^{(p)}) - Q(\mathbf{w}^{(p)}, \mathbf{w}^{(p)}) \leq 0$ *for all* p.

Then the sequence $\mathbf{w}^{(p)}$ *converges to some* $\mathbf{w}^*$ *in the closure of* Ω.

Proof. Using Lemma 4.1, the identity $J(\mathbf{w}) = Q(\mathbf{w}, \mathbf{w})$ and the definition of $\mathbf{w}^{(p)}$, we can derive the following inequality:

$$\begin{aligned} J(\mathbf{w}^{(p+1)}) - J(\mathbf{w}^{(p)}) &= J(\mathbf{w}^{(p+1)}) - Q(\mathbf{w}^{(p+1)}, \mathbf{w}^{(p)}) + Q(\mathbf{w}^{(p+1)}, \mathbf{w}^{(p)}) - Q(\mathbf{w}^{(p)}, \mathbf{w}^{(p)}) \\ &\leq Q(\mathbf{w}^{(p+1)}, \mathbf{w}^{(p)}) - Q(\mathbf{w}^{(p)}, \mathbf{w}^{(p)}) \leq 0, \end{aligned}$$

as required to prove the convergence of $\mathbf{w}^{(p)}$ to some $\mathbf{w}^*$. □

4.2 Decomposing the optimization problem

The dual QP given by Eq. (4.10) can be solved using standard QP techniques. However, the number of variables α in Eq. (4.10) are exponential in the length of observation sequence T. Therefore, converting the dual QP given by Eq. (4.10) into a standard QP form yields a representation that employs a matrix of size $KNM^T \times KNM^T$ (M the number of states), which leads to a very large scale problem in general. We now introduce a simple, memory efficient algorithm for solving the dual QP by decomposing it into small problems.

The core idea of our algorithm is based on the fact that the constraints of Eq. (4.10) can be separated into N disjoint sets with respect to each training sample. This allows us to perform the search in rounds. On each round the

algorithm chooses a sample p and optimizes the Lagrange multipliers related to this sample.

Let us define a vector $\boldsymbol{\alpha}_{i,k}$ which is formed by simply stacking all the variables $\alpha_{i,k,\mathbf{q}}$ together for the sample i associated to category k, and the corresponding matrix $\mathbf{K}_{i,j,k}$ whose elements are the values of the kernel function in accordance with the Eq. (4.10). Using this notation we can rewrite the Eq. (4.10) in the following vector form:

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{Q}(\alpha) = \frac{1}{2}\sum_{k}\sum_{i,j}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,j,k}\boldsymbol{\alpha}_{j,k} - \sum_{k}\sum_{i}\delta_{y_i,k}\boldsymbol{\alpha}_{i,k}^{T}\bar{\mathbf{1}} \\ \texttt{s.t.} \quad & \forall i,k \quad \boldsymbol{\alpha}_{i,k} \leq C\bar{\mathbf{1}}_{\hat{\mathbf{q}}_i};\ \forall i \quad \sum_{k}\boldsymbol{\alpha}_{i,k}^{T}\bar{\mathbf{1}} = 0. \end{aligned} \tag{4.12}$$

Let us fix a sample index p and write the QP only in terms of the variables $\boldsymbol{\alpha}_{i,k}$. We now isolate the contribution of $\boldsymbol{\alpha}_{i,k}$ in $\mathcal{Q}$:

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\alpha}_{p,k}) &= \frac{1}{2}\sum_{k}\sum_{i,j}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,j,k}\boldsymbol{\alpha}_{j,k} - \sum_{k}\sum_{i}\delta_{y_i,k}\boldsymbol{\alpha}_{i,k}^{T}\bar{\mathbf{1}} \\ &= \frac{1}{2}\sum_{k}\boldsymbol{\alpha}_{p,k}^{T}\mathbf{K}_{p,p,k}\boldsymbol{\alpha}_{p,k} + \sum_{k}\sum_{i\neq p}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,p,k}\boldsymbol{\alpha}_{p,k} \\ &\quad + \frac{1}{2}\sum_{k}\sum_{i\neq p,j\neq p}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,j,k}\boldsymbol{\alpha}_{j,k} - \sum_{k}\delta_{y_p,k}\boldsymbol{\alpha}_{p,k}^{T}\bar{\mathbf{1}} - \sum_{k}\sum_{i\neq p}\delta_{y_i,k}\boldsymbol{\alpha}_{i,k}^{T}\bar{\mathbf{1}} \\ &= \frac{1}{2}\sum_{k}\boldsymbol{\alpha}_{p,k}^{T}\mathbf{K}_{p,p,k}\boldsymbol{\alpha}_{p,k} + \sum_{k}\left[\sum_{i\neq p}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,p,k} - \delta_{y_p,k}\bar{\mathbf{1}}^{T}\right]\boldsymbol{\alpha}_{p,k} \\ &\quad + \sum_{k}\left[\frac{1}{2}\sum_{i\neq p,j\neq p}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,j,k}\boldsymbol{\alpha}_{j,k} - \sum_{i\neq p}\delta_{y_i,k}\boldsymbol{\alpha}_{i,k}^{T}\bar{\mathbf{1}}\right]. \end{aligned} \tag{4.13}$$

By Defining $\mathbf{A}_{p,k} = \sum_{i\neq p}\boldsymbol{\alpha}_{i,k}^{T}\mathbf{K}_{i,p,k} - \delta_{y_p,k}\bar{\mathbf{1}}^{T}$ and omitting all the constants that do not affect the solution, we have

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{Q}(\boldsymbol{\alpha}_{\boldsymbol{p,k}}) = \frac{1}{2}\sum_{k}\boldsymbol{\alpha}_{p,k}^{T}\mathbf{K}_{p,p,k}\boldsymbol{\alpha}_{p,k} + \sum_{k}\mathbf{A}_{p,k}\boldsymbol{\alpha}_{p,k} \\ \texttt{s.t.} \quad & \forall k \quad \boldsymbol{\alpha}_{p,k} \leq C\bar{\mathbf{1}}_{\hat{\mathbf{q}}_p};\ \sum_{k}\boldsymbol{\alpha}_{p,k}^{T}\bar{\mathbf{1}} = 0. \end{aligned} \tag{4.14}$$

Note that the normalization constraints on the multipliers α are local to each example i. This allows us to perform a block-coordinate ascent where a block corresponds to the vector of multipliers α_i associated with a single example i. The general skeleton of the block-coordinate ascent algorithm is given in Figure 4.1. The algorithm is initialized with setting the multipliers $\alpha_{i,k,\mathbf{q}} = 0$ for all $\{i, k, \mathbf{q}\}$. At each iteration, we choose an example from the training set and improve the multipliers of this example by solving the isolated QP given in Eq. (4.14). The loop continues iterating as long as the algorithm does not meet a stopping criterion. A naive criterion is to run the algorithm for a fixed number of rounds. A better way which we discuss in the section 4.3 is to keep on running the loop until a predefined accuracy is not met.

To complete the details of the algorithm we need to address two issues. The first problem we have to solve is to design a scheme for choosing the sample p on each round for optimization. Two commonly used methods are to scan the training set sequentially or to choose a sample uniformly at random. In the following section we present a scheme for choosing a sample in a parsimonious manner. This scheme appears to perform better empirically than other naive schemes. Second, we need to discuss how to solve efficiently the isolated QP given by Eq. (4.14). A sequential minimum optimization approach is described in section 4.4 to address this issue. This method is more efficient than using the standard QP techniques, especially when it suffices to find an approximation to the optimal solution.

- **Input** $\{(\mathbf{x}_1, \hat{\mathbf{q}}_1, y_1), \cdots, (\mathbf{x}_N, \hat{\mathbf{q}}_N, y_N)\}$.
- **Set** $\alpha_{i,k,\mathbf{q}} = 0, \quad \forall i, k, \mathbf{q}$.
- **Main loop:**
  1. Choose a sample p.
  2. Compute the constants for the isolated optimization problem:
  $$\mathbf{A}_{p,k} = \sum_{i \neq p} \boldsymbol{\alpha}_{\boldsymbol{i,k}}{}^T \mathbf{K}_{i,p,k} - \delta_{y_p,k} \bar{\mathbf{1}}^T.$$
  3. Find $\boldsymbol{\alpha}_{p,k}$ to be the solution of the the reduced problem for one sample:
  $$\begin{aligned} \min_{\alpha} \quad & \mathcal{Q}(\boldsymbol{\alpha}_{\boldsymbol{p,k}}) = \frac{1}{2} \sum_k \boldsymbol{\alpha}_{p,k}^T \mathbf{K}_{p,p,k} \boldsymbol{\alpha}_{p,k} + \sum_k \mathbf{A}_{p,k} \boldsymbol{\alpha}_{p,k} \\ \texttt{s.t.} \quad & \forall k \quad \boldsymbol{\alpha}_{p,k} \leq C \bar{\mathbf{1}}_{\hat{\mathbf{q}}_p}; \ \sum_k \boldsymbol{\alpha}_{p,k}^T \bar{\mathbf{1}} = 0. \end{aligned}$$

Figure 4.1: Skeleton of the algorithm for learning kernel based hidden Markov model

4.3 Sample selection strategy

In this section we address the strategy of choosing a sample from the training set for isolated optimization, and the stopping criterion for outer loop is also presented.

According to optimization theory, the solutions for the optimization problem given by Eq. (4.14) must satisfy the KKT conditions. Therefore, we can choose those samples who do not meet the KKT conditions. Before deriving the formal criterion, let us recall the Lagrangian of the problem, that is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, u, \theta) = \frac{1}{2}\sum_{k} \boldsymbol{\alpha}_{p,k}^{T}\mathbf{K}_{p,p,k}\boldsymbol{\alpha}_{p,k} + \sum_{k}\mathbf{A}_{p,k}\boldsymbol{\alpha}_{p,k} \\ - \sum_{k}\bar{\mathbf{u}}_{k}^{T}\left(C\bar{\mathbf{1}}_{\hat{\mathbf{q}}_i} - \boldsymbol{\alpha}_{p,k}\right) - \theta\sum_{k}\boldsymbol{\alpha}_{p,k}^{T}\bar{\mathbf{1}}\end{aligned} \tag{4.15}$$

The first condition is,

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}_{p,k}} = \mathbf{K}_{p,p,k}\boldsymbol{\alpha}_{p,k} + \mathbf{A}_{p,k}^{T} + \bar{\mathbf{u}}_{k} - \theta\bar{\mathbf{1}} = \bar{\mathbf{0}}. \tag{4.16}$$

Define $F(p,k,\mathbf{q}) = \sum_i \sum_{\mathbf{q}'} \alpha_{i,k,\mathbf{q}'} K(\mathbf{q}, \mathbf{x}_p, \mathbf{q}', \mathbf{x}_i) - \delta_{y_p,k}$. Using this definition and KKT conditions, we get the following set of constraints on a feasible solution for the isolated QP:

$$\forall k, \mathbf{q} \qquad F(p,k,\mathbf{q}) + u_{k,\mathbf{q}} = \theta; \tag{4.17}$$

$$\forall k, \mathbf{q} \qquad u_{k,\mathbf{q}}\left(C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} - \alpha_{p,k,\mathbf{q}}\right) = 0; \tag{4.18}$$

$$\forall k, \mathbf{q} \qquad u_{k,\mathbf{q}} \geq 0. \tag{4.19}$$

We can further simplify the equations by considering two cases. The first case is when there is a $\alpha_{p,k,\mathbf{q}}$ such that $\alpha_{p,k,\mathbf{q}} = C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}}$. In this case Eq. (4.18) holds automatically. By combining Eq. (4.17) and Eq. (4.19) we have,

$$F(p,k,\mathbf{q}) \leq \theta. \tag{4.20}$$

In the second case $\alpha_{p,k,\mathbf{q}} < C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}}$ for each element of $\mathbf{q}$. In order for Eq. (4.18) to hold we must have $u_{k,\mathbf{q}} = 0$. Thus, using Eq. (4.17) we get that,

$$F(p,k,\mathbf{q}) = \theta. \tag{4.21}$$

As stated before, the constraints on α from the isolated optimization problem given by (4.14) imply that for all k, $\boldsymbol{\alpha}_{p,k} \leq C\bar{\mathbf{1}}_{\hat{\mathbf{q}}_p}$, and $\sum_k \boldsymbol{\alpha}_{p,k}^T \bar{\mathbf{1}} = 0$. As such, if these constraints are satisfied there must exist at least one variable $\alpha_{p,k,\mathbf{q}}$ such that $\alpha_{p,k,\mathbf{q}} < C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}}$. Let us define $\Lambda = \{k, \mathbf{q} : \alpha_{p,k,\mathbf{q}} < C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}}\}$. We thus get that $\theta = \min_{\{k,\mathbf{q}\}\in\Lambda} F(p,k,\mathbf{q})$. Finally, we obtain:

$$\max_{k,\mathbf{q}} F(p,k,\mathbf{q}) \leq \min_{\{k,\mathbf{q}\}\in\Lambda} F(p,k,\mathbf{q}). \tag{4.22}$$

We now define the difference as follows:

$$d_p = \max_{k,\mathbf{q}} F(p,k,\mathbf{q}) - \min_{\{k,\mathbf{q}\}\in\Lambda} F(p,k,\mathbf{q}). \tag{4.23}$$

According to the definition, $F(p,k,\mathbf{q})$ is the confidence of input signal $\mathbf{x}_p$ along the state path $\mathbf{q}$ for the model k. As such, we can compute the maximum of $F(p,k,\mathbf{q})$ for all p and k through our inference algorithm presented in the previous chapter. Similarly, the right item of inequality above can be calculated by adjusting the inference algorithm to find the minimum over the path sequence space Λq.

Since $\max_{k,\mathbf{q}} F(p,k,\mathbf{q}) \geq \min_{\{k,\mathbf{q}\}\in\Lambda} F(p,k,\mathbf{q})$ then necessary condition to be an optimum for Eq. (4.14) is that $d_p = 0$. In the actual numerical implementation, we will relax this condition to a given tolerance $0 \leq \epsilon \ll 1$ such that $d_p \leq \epsilon$. Therefore, we keep performing the main loop of Figure 4.1 so long as there are examples $(\mathbf{x}_p, y_p)$ whose values d_p are greater than ϵ.

The variables d_p also serve as our criterion for selecting an sample for an update. In our implementation we select the sample index p for which the associated d_p

is maximal. We then find the multipliers $\alpha_{p,k,\mathbf{q}}$ for all possible $k, \mathbf{q}$ which are the solution of the isolated optimization problem given in Eq. (4.14). Due to the change in $\alpha_{p,k,\mathbf{q}}$ we need to recalculate the maximum of $F(p, k, \mathbf{q})$ and the minimum of $F(p, k, \mathbf{q})$ over Λ for all p and k. The pseudo-code describing this process is deferred to the next section in which we present an efficient algorithm for solving the isolated QP problem without any extra matrix storage and without using numerical QP optimization steps at all.

4.4 Sequential minimal optimization

In the case of the problems with a small number of classes k, sequences T and states M, the standard QP techniques can solve efficiently the isolated dual problems above. However, this could be intractable when we are facing the larger size of the isolated dual problem. Fortunately, we do not need to solve the reduced optimization problem above at each pass through the data, but just involving two Lagrange multipliers at each optimization step. It is analogous to the sequential minimal optimization (SMO) method used in SVM [Pla99].

The core idea of Sequential Minimal Optimization (SMO) approach is to take an ascent step that modifies the least number of variables.In our case, we must alter simultaneously at least two Lagrange multipliers, in order to respect the normalization constraint given by Eq. (4.14). This can be satisfied by moving weight from one dual variable to another.

In the following subsections, we address two components crucial to this optimization problem, an analytic solution for optimizing two multipliers simultaneously and a strategy to selecting these two multipliers.

4.4.1 Optimizing two multipliers

Assume we have picked two multipliers from sample 1 and denote these two multipliers as α_1 and α_2, without loss of generality. Similarly, the state sequences associated to these two multipliers are represented to $\mathbf{q}_1$ and $\mathbf{q}_2$. We also let the new values of these two multipliers be α_1^* and α_2^*, respectively. Due to the normalization constraint, we define a constant D and have the following equality:

$$D = \alpha_1^* + \alpha_2^* = \alpha_1 + \alpha_2$$

We can simplify Eq. (4.14), by considering the two cases. The first case is when both α_1 and α_2 belong to the same category, called 1. In this case, those multipliers from other categories do not affect the solution and can be omitted from the optimization problem. We have

$$\begin{aligned}\mathcal{Q}'(\alpha_1,\alpha_2) =& \frac{1}{2}K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_1,\mathbf{x}_1)\alpha_1^2 + \frac{1}{2}K(\mathbf{q}_2,\mathbf{x}_1,\mathbf{q}_2,\mathbf{x}_1)\alpha_2^2 + K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_2,\mathbf{x}_2)\alpha_1\alpha_2 \\ &+ \alpha_1\sum_{s=3}^{S}\alpha_s K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_1) + \alpha_2\sum_{s=3}^{S}\alpha_s K(\mathbf{q}_2,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_1) \\ &+ \alpha_1\sum_{i=2}^{N}\sum_{s=1}^{S}\alpha_s K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_i) + \alpha_2\sum_{i=2}^{N}\sum_{s=1}^{S}\alpha_s K(\mathbf{q}_2,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_i) \\ &- \delta_{y_1,1}\alpha_1 - \delta_{y_1,1}\alpha_2,\end{aligned}$$

where S is the number of all the possible state sequences in the state space $\mathcal{Q}$. With slight abuse of notation, we replace $\mathcal{Q}'$ as $\mathcal{Q}$ and have the following definitions:

$$V_1 = \sum_{s=3}^{S}\alpha_s K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_1) \qquad V_2 = \sum_{s=3}^{S}\alpha_s K(\mathbf{q}_2,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_1)$$

$$\theta_1 = \sum_{i=2}^{N}\sum_{s=1}^{S}\alpha_s K(\mathbf{q}_1,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_i) \qquad \theta_2 = \sum_{i=2}^{N}\sum_{s=1}^{S}\alpha_s K(\mathbf{q}_2,\mathbf{x}_1,\mathbf{q}_s,\mathbf{x}_i).$$

Along the linear the linear equality constraint and the above definition, the objective function $\mathcal{Q}$ can be expressed in terms on α_2 alone:

$$\begin{aligned}\mathcal{Q}(\alpha_2) =& \frac{1}{2}K_{11}(D-\alpha_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + K_{12}(D-\alpha_2)\alpha_2 \\ &+ (V_1+\theta_1-\delta_{y_1,1})(D-\alpha_2) + (V_2+\theta_2-\delta_{y_1,1})\alpha_2,\end{aligned} \tag{4.24}$$

where $K_{11} = K(\mathbf{q}_1, \mathbf{x}_1, \mathbf{q}_1, \mathbf{x}_1)$, $K_{22} = K(\mathbf{q}_2, \mathbf{x}_1, \mathbf{q}_2, \mathbf{x}_1)$ and $K_{12} = K(\mathbf{q}_1, \mathbf{x}_1, \mathbf{q}_2, \mathbf{x}_1)$. The extremum of the object function is at

$$\frac{d\mathcal{Q}}{d\alpha_2} = -K_{11}(D-\alpha_2)+K_{22}\alpha_2+K_{12}D-2K_{12}\alpha_2-(V_1+\theta_1-\delta_{y_1,1})+(V_2+\theta_2-\delta_{y_1,1}) = 0.$$

If the second derivative is positive, which is the usual case, then the minimum of α_2 can be expressed as

$$\alpha_2^*(K_{11}+K_{22}-2K_{12}) = (K_{11}-K_{12})D + (V_1+\theta_1) - (V_2+\theta_2).$$

According the definition of $F(p, k, \mathbf{q}$, for the state sequences $\mathbf{q}_1$ and $\mathbf{q}_2$ we have F_1 and F_2 as follows:

$$F_1 = F(1, 1, \mathbf{q}_1) = K_{11}\alpha_1 + K_{12}\alpha_2 + V_1 + \theta_1 - \delta_{y_1,1}$$

$$F_2 = F(1, 1, \mathbf{q}_2) = K_{12}\alpha_1 + K_{22}\alpha_2 + V_2 + \theta_2 - \delta_{y_1,1}$$

Therefore, we compute the minimum of multiplier α_2 through

$$\alpha_2^* = \alpha_2 + \frac{F_1 - F_2}{K_{11}+K_{22}-2K_{12}}.$$

In the second case that α_1 and α_2 belong to different categories (suppose 1 and 2), we use the similar derivation and get the solution for finding the minimum of multiplier α_2 through

$$\alpha_2^* = \alpha_2 + \frac{F_1 - F_2 + \delta_{y_1,1} - \delta_{y_1,2}}{K_{11}+K_{22}}.$$

Note that we can rewrite the new assignments of the given multipliers in these two cases to a unified form, as

$$\begin{aligned} \alpha_2^* &= \alpha_2 + \frac{(F_1 + \delta_{y_1,k_1}) - (F_2 + \delta_{y_1,k_2})}{K_{11} + K_{22} - 2\delta_{k_1,k_2}K_{12}} \\ \alpha_1^* &= \alpha_1 - \frac{(F_1 + \delta_{y_1,k_1}) - (F_2 + \delta_{y_1,k_2})}{K_{11} + K_{22} - 2\delta_{k_1,k_2}K_{12}}, \end{aligned} \tag{4.25}$$

where k_1, k_2 are the class label associated with α_1 and α_2, respectively.

In addition to the normalization constraints, the update of α also need to meet the marginal constraints $\alpha_{p,k,\mathbf{q}} \leq C\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}}$. Considering together with normalization constraints, we derive the bound of the multipliers α as

$$\begin{aligned} 0 \leq \alpha_{p,k,\mathbf{q}} \leq C \qquad &\text{if} \quad \delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} = 1; \\ -C \leq \alpha_{p,k,\mathbf{q}} \leq 0 \qquad &\text{if} \quad \delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} = 0. \end{aligned}$$

In the Figure 4.2 we show an example for optimizing a multiplier with which $\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} = 1$ is associated. It is clear that the above bounds cause the Lagrange multipliers to lie within the bounds. The updated optimum either occurs at the minimum of the parabola if it is feasible or the upper or lower bound otherwise, as shown in the figure. The similar situation is hold when $\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} = 0$.

4.4.2 Selecting SMO pairs

In the previous subsection we present an algorithm for optimizing the two chosen multipliers α_1 and α_2. As long as the algorithm always optimizes and updates two multipliers at every step, then each step will decrease the objective function. The optimum will be found when no one Lagrange multiplier violate the KKT conditions. Note that KKT conditions are also used as the strategy to select the sample in the section 4.3. To remind the reader, we choose the sample who have

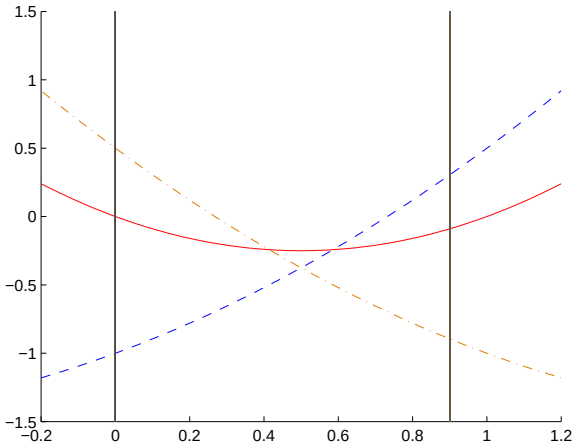


Figure 4.2: The bound of optimum of a multiplier with $\delta_{y_p,k}\delta_{\hat{\mathbf{q}}_p,\mathbf{q}} = 1$, horizontal axis represents α with two vertical lines depicting the lower and upper bounds 0 and C, respectively. Vertical axis represents the objective cost. Optimum either occurs at the minimum of the parabola if it is feasible or the lower or upper bound otherwise.

the maximal $d_p = \max_{k,\mathbf{q}} F(p,k,\mathbf{q}) - \min_{\{k,\mathbf{q}\}\in\Lambda} F(p,k,\mathbf{q})$. As long as there is a multiplier who violates the KKT conditions, the variable d of the corresponding sample must be greater than zero, and eventually the sample would be chosen for optimization. The computation of d is determined by two state sequences, the most likely state sequence for input signal and the least likely state sequence among the support vectors associated with the input signal. They all must violate the KKT conditions. Therefore, we can choose their multipliers for optimization and keep performing the selection until the variable d corresponding to the given sample is less than the predefined tolerance ϵ.

We are now ready to describe the complete implementation of the algorithm for learning kernel based hidden Markov model. The algorithm gets a required accuracy parameter ϵ, ϵ' and the value of C. We begin the algorithm with $\alpha_{i,k,\mathbf{q}} = 0$ for all indices and the initial estimates of the hidden state sequence. Such initial estimates can be obtained in a number of ways, for example, manual segmentation of the input sequences into states and segmental k-means segmentation with clustering. After the initial estimates, the algorithm goes to main loop for iteratively reestimating the weights of the model using two-step learning algorithm. On each iteration we compute from $F_{i,1}$ and $F_{i,2}$ the value d_i for each sample and choose the sample index p for which d_p is the largest. We then call the sequential minimal optimization algorithm which in turn finds the solution to the isolated QP problem for the sample indexed p. This algorithm selects two Lagrange multipliers who violate the KKT conditions and find the optima of these two multipliers. This process is repeated so long as the value d_i is larger than ϵ for all $1 \leq i \leq N$. We outline the complete algorithm in Figure 4.3.

Input $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.
Initialize

1. $\alpha_{i,k,\mathbf{q}} = 0 \quad \text{for} i = 1, \ldots, N, k = 1, \ldots, K, \mathbf{q} \in \mathcal{Q}$
2. $\hat{\mathbf{q}}_i \; i = 1, \ldots, N$

Repeat

- Initialize for $i = 1, \ldots, N, k = 1, \ldots, K, \mathbf{q} \in \mathcal{Q}$
  1. $F_{i,1} = \max_{k,\mathbf{q}} F(p, k, \mathbf{q}) \quad (k_{i,1}, \mathbf{q}_{i,1}) = \arg\max_{k,\mathbf{q}} F(p, k, \mathbf{q})$
  2. $F_{i,2} = \min_{\{k,\mathbf{q} \in \Lambda\}} F(p, k, \mathbf{q}) \quad (k_{i,2}, \mathbf{q}_{i,2}) = \arg\min_{\{k,\mathbf{q} \in \Lambda\}} F(p, k, \mathbf{q})$
  3. $d_i = F_{i,1} - F_{i,2}$
- Repeat:
  - Set: $p = \arg\max_i d_i$
  - Repeat:
    - Set $k_1 = k_{p,1}$, $k_2 = k_{p,2}$, $\mathbf{q}_1 = \mathbf{q}_{p,1}$, $\mathbf{q}_2 = \mathbf{q}_{p,2}$
    - Update $\alpha_{p,k_1,\mathbf{q}_1}$ and $\alpha_{p,k_{p,1},\mathbf{q}_2}$ by the sequential maximal optimization algorithm
    - Update $F_{p,1}, F_{p,2}, d_p$
  - Until $d_p \leq \epsilon$
  - Update: $F_{i,1}, k_{i,1}, \mathbf{q}_{i,1} \quad \text{for } i = 1, \ldots, N$
  - Update: $F_{i,2}, k_{i,2}, \mathbf{q}_{i,2} \quad \text{for } i = 1, \ldots, N$
  - Update: $d_i \quad \text{for } i = 1, \ldots, N$
- Until $d_i \leq \epsilon \quad \text{for } i = 1, \ldots, N$

Until $\left|\sum_k \left(\|\mathbf{w}_k^*\|_2^2 - \|\mathbf{w}_k\|_2^2\right)\right| \leq \epsilon'$
Output:

$$h(\mathbf{x}) = \arg\max_k \left\{ \max_{\mathbf{q} \in \mathcal{Q}|_k} \left[\sum_{i, \mathbf{q}' \in \mathcal{Q}|_k} \alpha_{i,k,\mathbf{q}} K(\mathbf{x}_i, \mathbf{q}', \mathbf{x}, \mathbf{q}) \right] \right\}.$$

Figure 4.3: The complete two-step learning algorithm

4.5 Experimental results

In this section, a classification experiment based on a set of synthetic sequence data were conducted to study the characteristics of our proposed learning algorithm, and the difference in classification performances as compared to traditional learning method.

In this experiment, we individually generate two classes of the synthetic data set using two different first-order hidden Markov models. Each model is a left-right model and consists of three states. Every state is modeled as a Gaussian mixture with two components. To evaluate the performance of our method, we define the states belong to the different classes are so similar that there are very big overlap between them. Fig. 4.4 shows a scatter plot of some synthetic data generated for the two classes.

We divide the dataset into 10 training subsets and one testing subset. The testing data set contains 1000 samples with 15 time sequences. To evaluate the generalization of the different methods, we randomly choose the training samples with different size. The accuracy results, summarized in Figure 4.5, are averages over the 10 folds. We implemented the HMM and our proposed kernel based hidden Markov model (referred to KHMM in figure 4.5). The parameters of HMM were trained by the traditional maximum likelihood learning algorithm, while our proposed model was trained by margin maximization. The kernel function used in our proposed model was the RBF kernel [Hay99].

Figure 4.5 shows two types of gains in accuracy. The use of kernels leads our margin-based method to achieve a very significant gain in accuracy over the

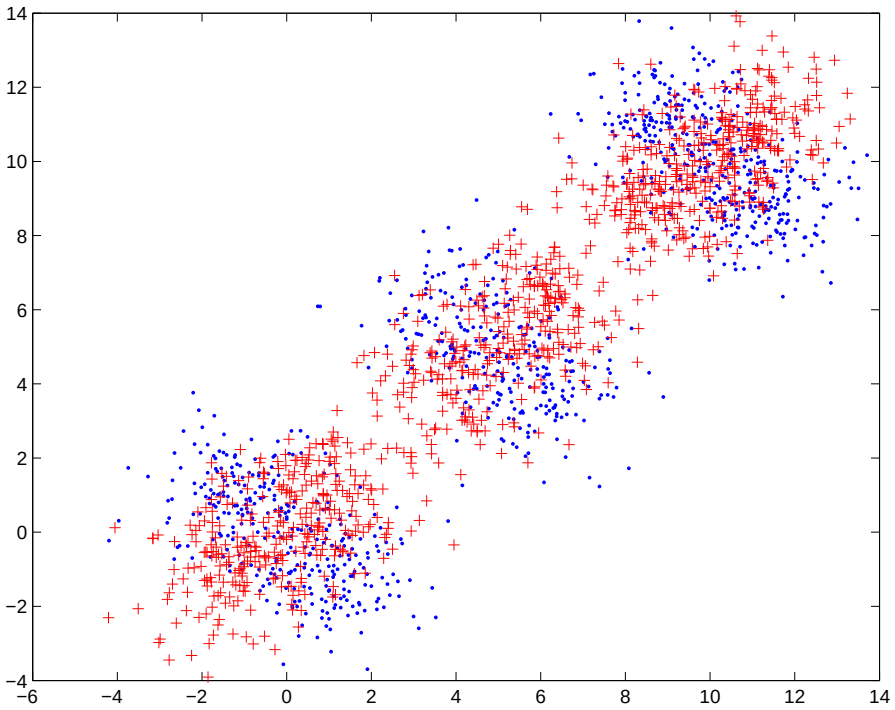


Figure 4.4: The distribution of synthetic data

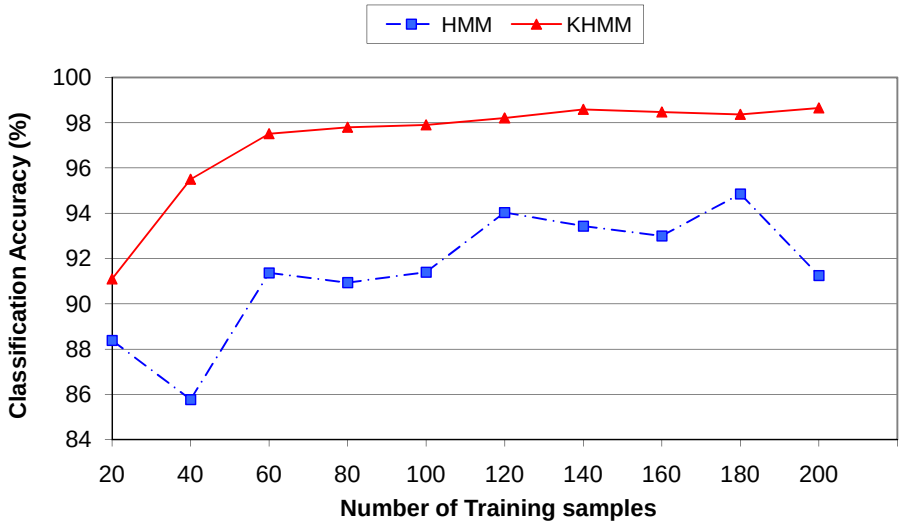


Figure 4.5: Average classification performance for HMM and KHMM

respective the likelihood maximizing method. Our approach gave 6.7% improvement in classification accuracy over ML method for 60 training samples, while 8.1$ improvement was achieved for 200 training samples. Furthermore, our proposed approach obtained a smoother curve with respect to the different size training sets. It shows that our method has better generalization performance in the case of the training data with big overlap. Interestingly, the traditional maximum likelihood method might cause the model to be overtrained when the training set is the size of 200, while the curve of the results given by our proposed model keep relatively smooth.

4.6 Conclusion

We presented here a kernel based hidden Markov model (KHMM) for classifying multi-class sequential signals/data. The model is capable of both exploiting the

temporal dynamics of the signals and maximizing the margins between classes in an effective way, by taking advantage of the rich language of hidden Markov model and superior separability of the kernel techniques. One of the most important contributions in our work is to propose a maximum margin discriminative learning method. It was presented with a two-step learning algorithm for constructing KHMM, a theoretical analysis showing its convergence to local minima, and complexity reduction method for optimization process.

The experimental results on the synthetic sequence data have shown that KHMM can exploit the nature of sequential signals and significantly outperforms the HMM based parametric methods. Overall, we believe that our proposed model will significantly further the applicability of high accuracy margin-based methods to real-world temporal signals. In the next chapter, we apply this framework to an application of brain computer interfaces, motor imagery signals.

Chapter 5

Motor imagery based brain computer interfaces

Improving classification accuracy is a key issue to advancing brain computer interface (BCI) research from laboratory to real world applications. This chapter presents a high accuracy EEG signal classification method using single trial EEG signal to detect left and right hand movement imagination. We apply an optimal temporal filter to remove irrelevant signal and subsequently extract key features from spatial patterns of EEG signal to perform classification. Specifically, the proposed method transforms the original EEG signal into a spatial pattern and applies the RBF feature selection method to generate robust feature. Classification is performed by the kernel based hidden Markov framework presented in the chapter 3, and our experimental results have shown significant improvement in classification accuracy over SVMs and HMMs.

5.1 Introduction

A brain-computer interface (BCI) is a communication system that does not depend on the brain's normal output pathways of peripheral nerves and muscles [WBM+02]. At present, eletroencephalography (EEG) is one of the most prevailing signals used in non-invasive BCI systems.

There are various kinds of EEG based BCIs categorized by the signals used [WBM+02]. Typical signals include slow cortical potential, μ/β rhythms, EEG (de)synchronization evoked by motor imagery, steady-state visual evoked potential, P300 potential, etc. EEG signals evoked by limb movement or motor imagery are of interest to this chapter.

The preparation, actual operation and mental imagination of limb movements activate similar EEG changes at sensorimotor areas on the scalp. When such regions become activated, EEG activities display an amplitude attenuation or event-related desynchronization (ERD). For instance, imagination of right-hand or left-hand movement results in the most prominent ERD localized over the corresponding sensorimotor cortex. However, ERD is subject-related, i.e. different subjects have different spatial localizations of ERD. This leads to difficulty when extracting features for classification.

Pfurtscheller et. al. [PNFP97] extracted motor imagery signals from $C3$ and $C4$ EEG Channels to build an online BCI system. The features presented to the classifier were short-term power spectra in pre-define frequency bands. This system using a LVQ algorithm achieved an accuracy of approximately 80% for 3 subjects.

Studies showed that the position of ERD may vary from subject to subject, and are not necessarily located beneath electrode positions $C3$ and $C4$ [PN01]. As

such, using more channels of signals may improve performance. Müller-Gerking et. al. [MGPF99] proposed to use Common Spatial Patterns (CSP) for the classification of motor execution or imagery signals. The CSP method resulted in significant improvement to performance as compared to their previous work in [PNFP97].

To extract the more significant and reliable features, as we mentioned before, the noises interfering with the interesting signals have to be reduced. Principal Component Analysis (PCA), as one of the popular noise reduction methods, has been widely used in statistical pattern recognition and signal processing. The main idea of PCA is to transform the data space to a feature space where the features are uncorrelated with each other. By retaining the features which have largest variances, the noises can be reduced to a certain degree. Motivated by this, in this chapter we attempt to develop a mathematical process to combine CSP feature extraction method with PCA method. The resulted transformation is equivalent to a set of spatial filters optimized to distinguish between the left and right hand movement or motor imagery.

In addition, temporal filtering was applied to reduce noise. In the past, the selection of frequency bands was limited to a few pre-defined bands [MGPF99, PN01]. In this chapter, we investigated the effects of temporal filtering for specific subject by an exhaustive search over all the frequency bands. We showed that classification performance could be improved significantly by applying proper band-pass filter.

To further enhance recognition accuracy, a Radial Basis Function (RBF) based feature selection and generation algorithm [CYB96] was adapted. We applied the Orthogonal Least Square (OLS) algorithm [CYB96] to feature selection and

generation. The extracted features are then used to train the SVMs, HMMs and our proposed KHMM framework. We show that our models significantly outperform other two approaches.

5.2 Experimental paradigm

In the standard paradigm for the discrimination of two mental states, the experimental task is to imagine either right-hand or left-hand movement depending on a visually presented cue stimulus. The subject was instructed to fixate on a computer screen about in $180cm$ front of him. Each trial was nine seconds long (Figure 5.1), starting with a blank screen which indicated a pause. At the 2nd second, an acoustic stimulus indicates the beginning of the trial, the blank screen was replaced by a cross "'+"' for $1s$. At $t = 3s$, a prompting arrow stimulus was displayed as cue, pointing either to the left or to the right lasting for 6 seconds. Following the direction of the arrow, the subject performed motor imagery accordingly.

Two male subjects (age 30-40 years) took part in the study, and both of them were not familiar with BCI. They are free from medication and central nervous system abnormalities. There were three experiments run in the different days, and two sessions of them were performed by subject A. The complete session consisted of five runs, each run consisted of 20 trials. The number of left and right hand imaginations are balanced.

EEG signals were recorded using the Neuroscan SynAmp2 system, sampled at 250 Hz. 28 channels of EEG around the C3 and C4 region related to the sensorimotor cortex were then chosen from the 64 scalp electrodes. EEG signals between 100 ms before stimuli and 4000 ms after stimuli were extracted for later

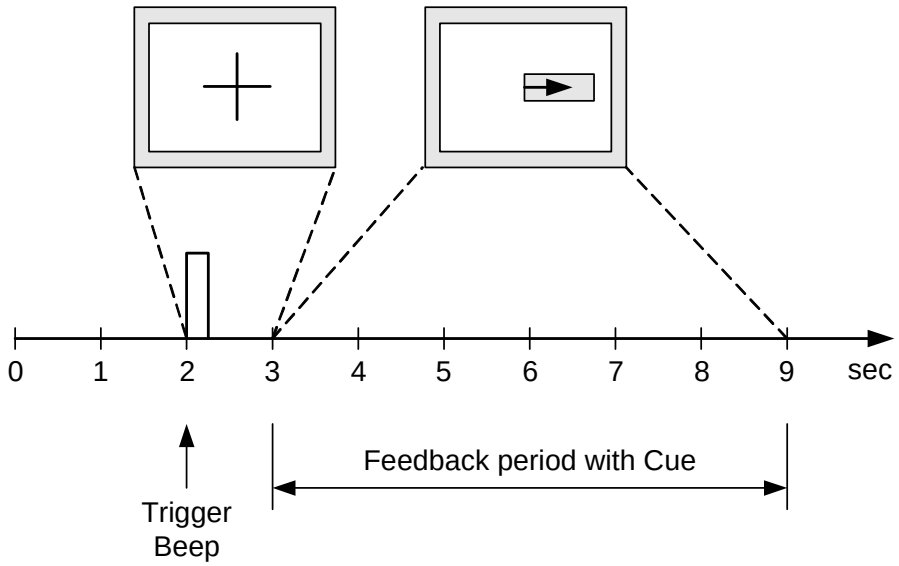


Figure 5.1: Timing scheme for the motor imagery experiments

processing.

5.3 EEG feature extraction

A classification task is usually broken up into two main parts. The first part is the extraction of relevant features that capture the class-invariant characteristics from some trial. The second part is the classification algorithm, performed on the extracted features. In this section, we present the feature extraction realized by projections of the high-dimensional, spatial-temporal raw signals onto very few specifically designed spatial filters. These filters are designed in such a way that the variances of the resulting signals carry the most discriminative information. The adjunct of the filters are called Common Spatial Patterns, and they are obtained from a set of calibration data by the method of CSP.

The features for the classification proper are vectors whose elements are the variances of the projected signals. These feature vectors are finally classified by machine learning approaches, such as simple linear Bayes classifiers, support vector machine,hidden Markov model, whose parameters a obtained again from the same set of calibration data, after projection onto the CSPs obtained in the first step.

The method of CSPs now finds a decomposition of the two groups of recordings into modes that are common to both groups, but maximally suited to distinguish between the groups. Mathematically, the method relies on the simultaneous diagonalization of two matrices closely related to the covariance matrices.

Given an N-channels spatial-temporal EEG signal $\mathbf{X}$, where $\mathbf{X}$ is a $N \times K$ matrix and K denotes the number of samples in each channel. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_K]$, the covariance matrix for i-th trial is

$$\mathbf{R}^{(i)} = \sum_{k=1}^{K} \left(\mathbf{x}_k^{(i)} - \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k^{(i)} \right) \left(\mathbf{x}_k^{(i)} - \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k^{(i)} \right)^T \tag{5.1}$$

where $\mathbf{x}_k^{(i)}$ is an N-dimensional vector at time k. This way, we can estimate covariance matrices for left and right hand data respectively. The normalized covariance matrices are

$$\mathbf{R}_L = \frac{1}{l} \sum_{i=1}^{l} \frac{\mathbf{R}_L^{(i)}}{trace(\mathbf{R}_L^{(i)})} \qquad \mathbf{R}_R = \frac{1}{r} \sum_{i=1}^{r} \frac{\mathbf{R}_R^{(i)}}{trace(\mathbf{R}_R^{(i)})} \tag{5.2}$$

where $\mathbf{R}_L$, $\mathbf{R}_R$ are the normalized covariance matrices and l, r denote numbers of trials, for the left and right hand data respectively.

The common spatial pattern [MGPF99] is extracted based on the simultaneous diagonalize of two covariance matrices belonging to left and right hand movement, and the resulted decomposition maximizes the differentiation between two groups

of data. After we have covariance matrices R_L and R_R, we can find

$$\mathbf{R} = \mathbf{R}_L + \mathbf{R}_R = \mathbf{U}\lambda\mathbf{U}^T \tag{5.3}$$

where $\mathbf{U}$ and λ are the eigenvectors and eigenvalues of $\mathbf{R}$ respectively. With these matrices, we can find a transformation matrix $\mathbf{W} = \lambda^{-\frac{1}{2}}\mathbf{U}^T$ to calculate CSP features. For details on this, refer to [MGPF99]. Here, we modify the approach of [MGPF99] by combining PCA, i.e., we only use the p principal eigenvectors from U to form the transformation matrix

$$\mathbf{W}_s = \lambda_s^{-\frac{1}{2}}\mathbf{U}_s^T \tag{5.4}$$

where $\mathbf{U}_s$ is composed of the p most significant eigenvectors of $\mathbf{U}$, $p \leq N$, and λ_s the corresponding eigenvalues. We can then evaluate the transformed covariance matrices $\mathbf{S}_L, \mathbf{S}_R$ as

$$\mathbf{S}_L = \mathbf{W_s}\mathbf{R}_L\mathbf{W_s}^T \quad \text{and} \quad \mathbf{S}_R = \mathbf{W_s}\mathbf{R}_R\mathbf{W_s}^T \tag{5.5}$$

Hence,

$$\begin{aligned}
\mathbf{S}_L + \mathbf{S}_R &= \mathbf{W_s}\mathbf{R}\mathbf{W_s}^T \\
&= \lambda_s^{-\frac{1}{2}}\mathbf{U}_s^T\mathbf{U}\lambda\mathbf{U}^T\mathbf{U}_s^T\lambda_s^{-\frac{1}{2}} \\
&= \lambda_s^{-\frac{1}{2}}\begin{bmatrix} \mathbf{I}_p & 0 \end{bmatrix}\begin{bmatrix} \mathbf{I}_p \\ 0 \end{bmatrix}\lambda_s^{-\frac{1}{2}} \\
&= \mathbf{I}_p
\end{aligned} \tag{5.6}$$

where $\mathbf{I}_p$ is a $p \times p$ identity matrix. The above equation shows that the CSP criterion is still satisfied when using the sub-matrix $\mathbf{W}_s$. From (5.6), it can be

found that $\mathbf{S}_L$ and $\mathbf{S}_R$ share a common eigenvectors matrix $\mathbf{B}$ such that

$$\mathbf{S}_L = \mathbf{B}\lambda_L\mathbf{B}^T \tag{5.7}$$

$$\mathbf{S}_R = \mathbf{B}\lambda_R\mathbf{B}^T \tag{5.8}$$

For each trial, the data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_K]$, is transformed to $\mathbf{Y}$ by

$$\mathbf{Y} = \mathbf{B}^T\mathbf{W_s}\mathbf{X} = \mathbf{P}\mathbf{X} \tag{5.9}$$

where the matrix $\mathbf{Y}$ is of size $p \times K$. This matrix is used to obtain the final features for classification by

$$f_j = log\left(\frac{var(\mathbf{y}_j)}{\sum_{k=1}^{m/2} var(\mathbf{y}_k)}\right) \qquad \begin{array}{c} j = 1, \cdots, \frac{m}{2} \\ (m \leq p) \end{array} \tag{5.10}$$

and

$$f_j = log\left(\frac{var(\mathbf{y}_{p-m+j})}{\sum_{k=p-\frac{m}{2}+1}^{p} var(\mathbf{y}_k)}\right) \qquad \begin{array}{c} j = \frac{m}{2}+1, \cdots, m \\ (m \leq p) \end{array} \tag{5.11}$$

where $\mathbf{y}_j$ is the j-th row of $\mathbf{Y}$ and $var(\mathbf{y}_j) = \mathbf{y}_j\mathbf{y}_j^T$ is the variance of $\mathbf{y}_j$. The optimal variable m and p are found experimentally. We denote the features generated by this method *PCA+CSP*, as the transformation matrix $\mathbf{W}_s$ is found based on the p most significant principal components.

5.4 Feature selection and generation

In the CSP method, the first $m/2$ features are evaluated using the first $m/2$ rows of $\mathbf{Y}$ and the last $m/2$ features use the corresponding last $m/2$ rows of $\mathbf{Y}$ [MGPF99].

To improve the feature selection strategy, we perform feature selection by the Orthogonal Least Square (OLS) algorithm. The OLS is an efficient implementation of the forward stepwise feature selection method [CCG91]. It selects the "important" regressors from an initial linear regression model sequentially. As the OLS algorithm can be implemented very efficiently, it can be applied to select models from very large initial systems.

Here we apply OLS to select features from the input feature vectors calculated from (5.10) and (5.11). Suppose that the input feature row vector for i-th trial is denoted by $\mathbf{f}^{(i)} = [f_1^{(i)}, \cdots, f_m^{(i)}]^T$ and constitutes the training set $\mathbf{F}$,

$$\mathbf{F} = \begin{bmatrix} f_1^{(1)} & \cdots & f_m^{(1)} \\ \vdots & \vdots & \vdots \\ f_1^{(Q)} & \cdots & f_m^{(Q)} \end{bmatrix} \tag{5.12}$$

where $\mathbf{F}$ consists of Q trials and m features per trial.

To improve robustness of features, we apply OLS to find a parsimonious selection of features from $\mathbf{F}$. Let $\mathbf{z}_i$ denotes i-th column of $\mathbf{F}$, the subset model found is $\widetilde{\mathbf{F}}$,

$$\mathrm{OLS}_1(\mathbf{F}) = \widetilde{\mathbf{F}} = [\widetilde{\mathbf{z}}_1, \cdots, \widetilde{\mathbf{z}}_{\widetilde{m}}] \qquad 0 < \widetilde{m} \leq m \tag{5.13}$$

where OLS_1 denotes the OLS function and $\{\widetilde{\mathbf{z}}_i\}$ are the features chosen from input features using OLS method.

In addition to feature selection, the OLS algorithm may be used to generate the new features based on the training set. Chng et. al. [CYB96] introduced an efficient adaptive model selection method based on the OLS algorithm. It first select a small subset RBF models from a large initial one and subsequently applies a local learning step to modify the selected node's parameters. Using simulation

results, they showed that the selected model's performance is improved, and that the pre-set values of the initial network become less critical. In this chapter, the same algorithm denoted by OLS_2 is applied to generate feature vectors for our classifier, namely

$$\mathrm{OLS}_2(\mathbf{F}) = [\widetilde{\mathbf{z}}_1, \cdots, \widetilde{\mathbf{z}}_{\widetilde{m}}, \widehat{\mathbf{z}}_1, \cdots, \widehat{\mathbf{z}}_{\widehat{m}}] \tag{5.14}$$

where $\{\widehat{\mathbf{z}}_i\}$ are the newly generated features. We denote features generated by this method as *PCA+CSP+OLS*$_2$

5.5 Experimental results

We evaluate our approach on the classification of EEG signals for motor imagery, to distinguish left and right hand movement imagination. The data sets used were denoted as $\{A1, A2, B1\}$.

5.5.1 temporal filtering

Due to the lack of training sample (100 samples for each session), we use a public dataset, courtesy of Müller and Curio[BCM02], to find the frenquency band of motor imagery signals. Undoubtedly, the frequency band obtained by the public dataset would not be the optimum values for the datasets used in our experiments. But the purpose of our experiments is mainly to compare the performance of classification algorithms. Deploying the parameters of the temporal filters obtained from other motor imagery dataset does not affect our justification.

An Infinite Impulse Response (IIR) band-pass filter is applied on the raw data before it is sent for feature extraction. To evaluate the effect of cut-off frequency,

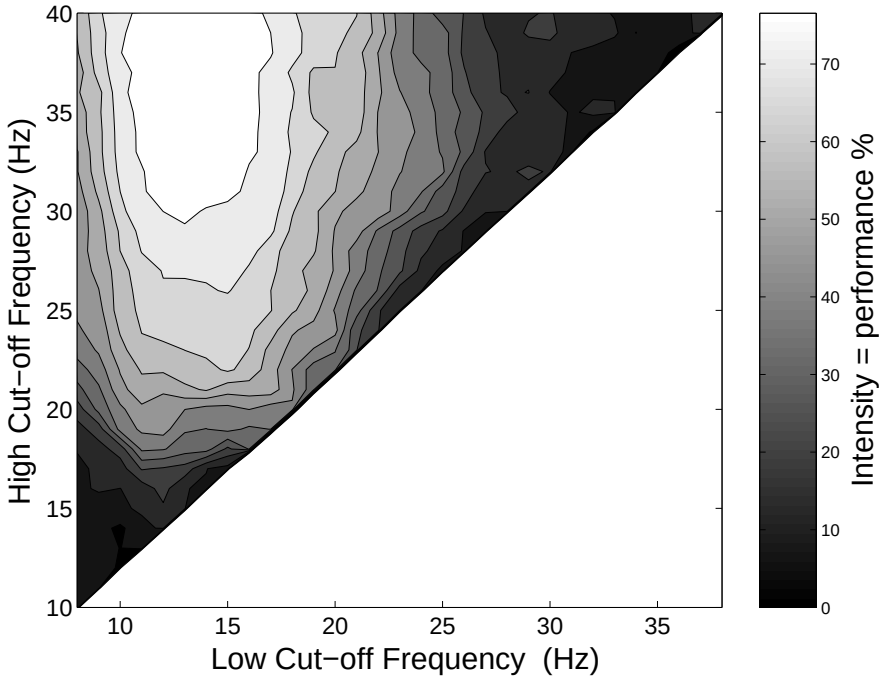


Figure 5.2: Evaluation Set: Classification accuracy using the different low/high cut-off frequency selection.

we perform an exhaustive search on various combinations of high and low cut-off frequencies by monitoring classification accuracy. Results are illustrated in Fig 5.2. The x-axis and y-axis of this figure are the low and high cut-off frequency of the bandpass filter respectively and the classification accuracy is reflected by the intensity level. It is found that when the low-cutoff frequency is in the range of $10-15$Hz and high cut-off frequency is about 30Hz, similar classification accuracy can be achieved.

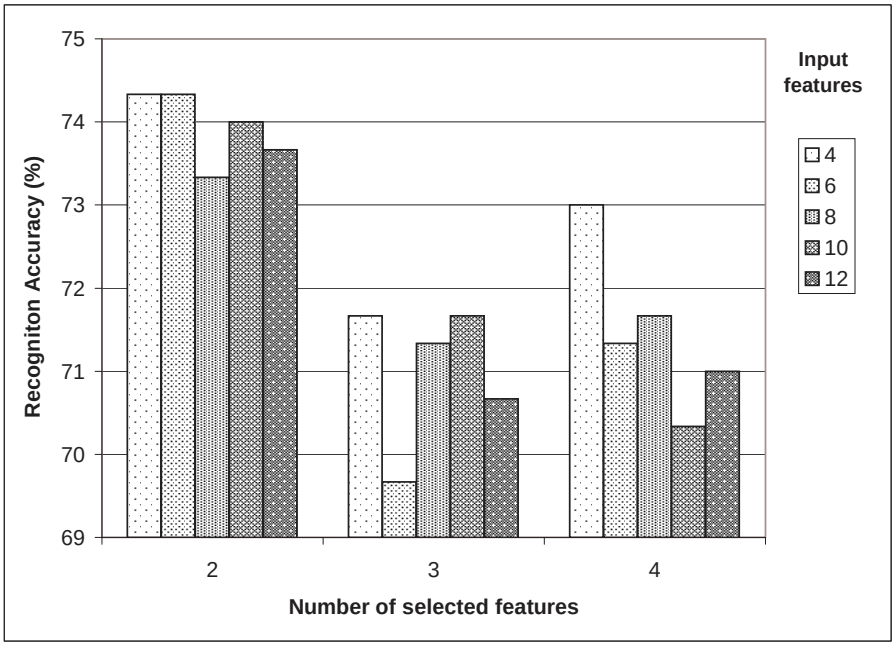


Figure 5.3: Evaluation Set: Classification performance using different number of features selected by OLS_1.

5.5.2 Optimization of Orthogonal Least Square Algorithm

The optimal parameter p for feature extraction is found to be 18 by similar approach described above. With this optimal p, feature extraction using (5.10) and (5.11) is performed and the OLS_1 is used to select features. Fig 5.3 shows the classification performance versus the number of features found by OLS_1. The best performance is obtained when $\tilde{m} = 2$ for various m and interestingly for $\tilde{m} = 2$, the selection is similar to what was used in basic CSP, i.e., the first and last row of $\mathbf{y}_j^i$ were used to evaluate f_1^i and f_2^i [MGPF99].

To further improve performance, the OLS_2 is applied to generate additional features. Fig 5.4 shows the classification performance on the evaluation set using

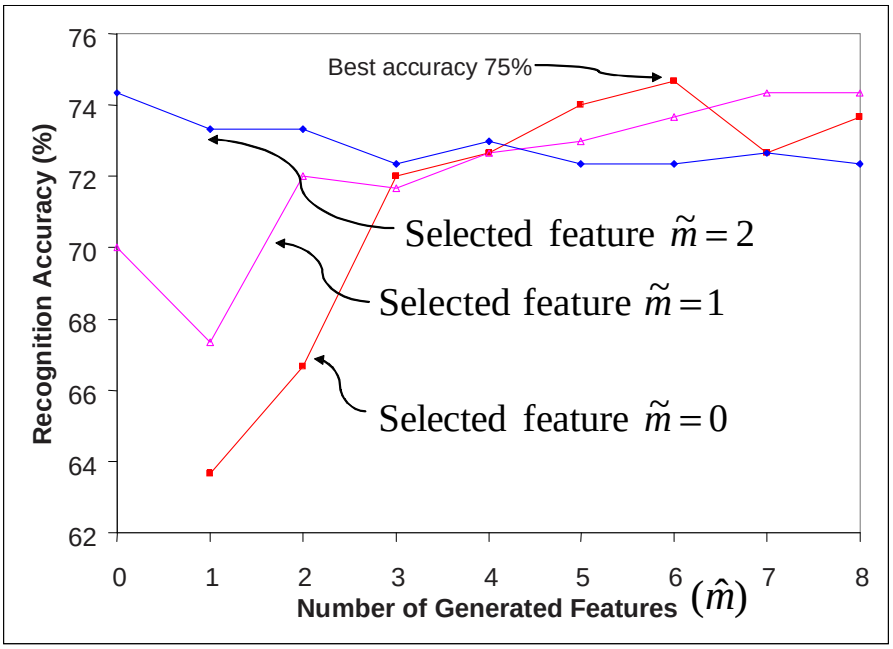


Figure 5.4: Evaluation Set: Classification performance using different number of selected and generated features obtained by OLS_2.

selected and generated features. Specifically, the x-axis on Fig 5.4 shows the number of generated features, and the different lines represent the experiments using 0, 1 and 2 selected features respectively. The best result is obtained for selected features $\tilde{m} = 0$ and generated features $\hat{m} = 6$.

5.5.3 Classification results

All of three datasets were divided into 20 folds of 95 training and 5 test samples each. Before classification, the time sequences are first divided into segments of $900ms$ length with $250ms$ overlap for feature extraction. For the purpose of comparison, common spatial patterns (CSP) features are employed in all classification methods. For each dataset, all of these $900ms$ long window signals were stacked

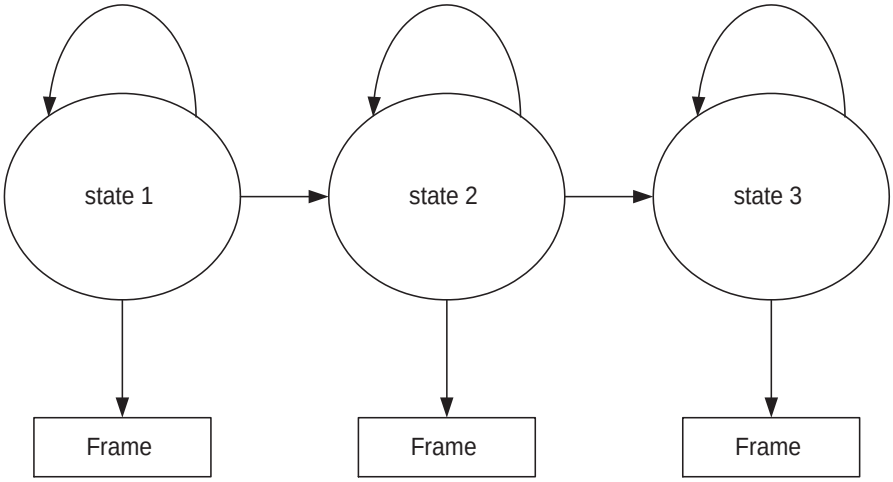


Figure 5.5: The state transition model used in the HMM and KHMM.

together to find the transformation matrix $\mathbf{W}_s$ for extracting features, as well as feature selection phase. Additionally, both HMM and our proposed method consist of 3 states (Figure 5.5) for capturing the structure of EEG data.

The kernel function used in SVM and KHMM is the RBF kernel [Vap98]. The classification results, shown in table 5.1, are averages over these 20 folds. We compare our proposed algorithm with other two classification approaches, SVM and HMM. In the A2 dataset, our proposed approach gave the highest classification accuracy of 93%, compared to the SVM (78%) and HMM (84%). For all of the datasets, KHMM obtained quite remarkable improvement over traditional likelihood maximization. Furthermore, the results shows that SVM always has the worst classification accuracy. The low classification accuracy may be due to the fact that it does not explicitly take the temporal dynamic of the signals into account.

	A1(%)	A2	B1
SVM(%)	70	78	72
HMM(%)	74	84	80
KHMM(%)	81	93	82
Improvement over SVM(%)	15.7	19.2	13.9
Improvement over HMM(%)	9.5	10.7	2.5

Table 5.1: Average classification performance for SVM, HMM and our proposed method.

5.6 Conclusion

In this chapter, we present our research result on the classification of EEG signal for distinguishing left and right hand movement, preparation or imagination. We propose to extract CSP features by combining PCA to reduce noise, and use OLS algorithm to generate features for classification. These methods are found effective and help achieve a high accuracy for single trial motor imagery classification.

The experimental results on real motor imagery EEG signal classification have shown that our proposed algorithm can exploit the nature of sequential signals and significantly outperforms the non-structural methods, and the HMM based parametric methods.

Chapter 6

Conclusion and future work

This thesis presents multiple aspects of brain computer interfaces for building a communication channel between brain and computer. We have shown that the performance or the transmit rate is largely dependent on the performance of classifiers for the brain signals. To address this issue, feature extraction and signal classification in brain signals based on graphical model, especially Markov random field, and the spatio-temporal characteristic are studied to deal with the heavily noisy, high deformable and non-stationary brain signal, such as eletroencephalography (EEG).

This work proposes a dynamic model referred to as kernel based hidden Markov model (KHMM) for classifying multi-class temporal signal data. This dynamic model incorporates kernel-based discriminative learning approaches into hidden Markov model, having no need of prior knowledge of signal distribution. The notion of Markov Random field is used to represent the interaction between signal observation and state, and the interaction between states as well. Given this MRF representation, the learning is formulated as finding the maximum margin

of separation between the category of the sample and the best runner-up in the kernel space. The formulation is by imposing the explicit constraint to the cost function so that the inferred state sequence from the designed model is the most possible state sequence. To effectively predict the brain's activities, a Viterbi dynamic programming is developed to recover all the state associated with the given observation sequence.

There are several theoretical advantages to our approach in addition to the empirical accuracy improvements we have shown experimentally. Because our approach only relies on using the maximum in the model for prediction, and does not require a normalized distribution $P(\mathbf{x}|y)$ over the model y, maximum margin estimation can be tractable when maximum likelihood is not. For example, to learn a probabilistic model $P(\mathbf{x}|y)$ over bipartite matchings using maximum likelihood requires computing the normalizing partition function, which is #P-complete. By contrast, maximum margin estimation can be formulated as a compact QP with linear constraints. Similar results hold for an important subclass of Markov networks and non-bipartite matchings.

This dissertation developed an efficient two-step learning algorithm for solving the training problem of the kernel based hidden Markov model. Because the underlying stochastic process is not usually observable and thus the optimal state sequence has to be estimated, the constrained optimization problem can not be solved directly using standard quadratic programming (QP) techniques. In the case of a partial or complete absence of the labels of states, the kernel based hidden Markov model suffers the chief computational bottleneck in learning the parameters of model. We solve this problem by alternatively estimating the parameters

of the designed model and the most possible state sequences, until convergence. It can be seen that this two-step algorithm is similar to the mathematics of standard Expectation-Maximization (EM) technique, although our optimization problem is not directly related to probability estimation.

In particular, an auxiliary function which averages over the values of the hidden variables given the parameters at the previous iteration is defined. By minimizing this auxiliary function, we will always carry out an improvement over the previous estimated parameters, unless finding the optimal values of parameters. The next step is to find a new parameter set of model which minimizes the constrained optimization problem given the previous estimated states sequence. To solve this optimization subproblem, we can use Karush-Kuhn-Tucker (KKT) theorem and the solution to the optimization problem is determined by the the saddle point of the function Q.

Although the second step in the two-step algorithm is a QP with linear number of variables and constraints in the size of the data, for most of real datasets, there would be thousands and tens of thousands possible states sequence and it is very difficult to solve using standard software. We present an efficient algorithm for solving the estimation problem called Structured SMO. Our online-style algorithm uses inference in the model and analytic updates to solve these extremely large estimation problems.

We then apply the kernel based hidden Markov model to the application of continuous motor imagery BCI system. In our framework, the user is just to imagine his/her hand movement and our system will execute the user's command depending on the prediction of which hand the user is imagining. This is guaranteed

by our developed high accuracy EEG signal classification algorithm which use single trial EEG signal to detect left and right hand movement imagination.

We first apply an optimal temporal filter to remove irrelevant signal and subsequently extract key features from spatial patterns of EEG signal to perform classification. The reason of employing multiple channel EEG signals is that the position of ERD may vary from subject to subject, and are not necessarily located beneath electrode positions $C3$ and $C4$. In addition, the noises interfering with the interesting signals would not be neglected if we build high performance BCI system. Therefore, in our framework a mathematical process to combine CSP feature extraction method with PCA method is developed. The resulted transformation is equivalent to a set of spatial filters optimized to distinguish between the left and right hand movement or motor imagery.

To further enhance recognition accuracy, a Radial Basis Function (RBF) based feature selection and generation algorithm was adapted. We applied the Orthogonal Least Square (OLS) algorithm to feature selection and generation. The extracted features are then used to train the SVMs, HMMs and our proposed KHMM framework. We show that our models significantly outperform other two approaches.

Our future work involves the theoretical analysis of the generalization bound of our proposed kernel based Markov model. As discussed above, our proposed dynamic model provides a minimum empirical risk owing to its maximum margin learning. However, how to relate the error rate on the training set to the generalization error is still an open question. This could be our future research. Moreover,

the study in the generalization bound of kernel based hidden Markov model is useful for theoretically comparing our proposed framework with other classification algorithm.

We have discussed so far the underlying principal and algorithmic issues that arise in the design of kernel based hidden Markov model. However, to make the proposed techniques practical in applications with large databases it will be beneficial for our future study to explore more technical improvements. These improvements would lead to a significant improvement in running time, while they do not change the underlying design principals. For example, in the section 4.3 we choose a sample to optimize the parameters of the model as long as its d_p is greater than a given tolerance ϵ ($0 \leq \epsilon \ll 1$). This may result in minuscule changes and a slow decrease in Q once most examples have been updated. To accelerate the process, especially on early iterations, a possible improvement is to use a variable tolerance, rather than a fixed accuracy. On early iterations the tolerance value is set to a high value so that the algorithm will spend only a small time on adjusting the weights of the support patterns. As the number of iterations increases we decrease ϵ and spend more time on adjusting the weights of support patterns.

Using the kernel based hidden Markov model to build a continuous motor imagery BCI system with high performance has been extensively studied in this thesis. However, this dynamic model can be applied to more types of BCI system. A possible candidate for such system is a continuous text input application. We may apply our proposed dynamic framework to the P3 word speller based on the P300 event related potential. Theoretically speaking, a kernel based hidden Markov model is capable of modeling stochastic processes of any length. In the case of word speller

application, however, it is desirable to model the characters that may be repeated in the longer continuous processes (word) using KHMM rather than modeling the continuous processes directly. How to connecting these character classifiers using the level-building strategy will be one of our future research directions.

We have presented a supervised learning framework for temporal signal classification with rich and interesting structure. Our approach has several theoretical and practical advantages over standard probabilistic models and estimation methods. We hope that continued research in this framework will help tackle evermore sophisticated classification problems in the future.

Bibliography

[BCD+94] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. Lecun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwriting digit recognition. In *International Conference On Pattern Recognition*, pages 77–87, 1994.

[BCM02] Benjamin Blankertz, Gabriel Curio, and Klaus-Robert Müller. Classifying single trial EEG: Towards brain computer interfacing. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, 2002.

[Ber95] D.P. Bertsekas. *Nonlinear Programming.* Athenas Scientific, Belmont, MA, 1995.

[BGH+99] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor. A spelling device for the paralysed. *Nature*, 398:297–298, March 1999.

[BS96] J.M. Belsh and P.L. Schiffman, editors. *Amyotrophic lateral sclerosis: diagnosis and management for the clinician*. Futura Publishing Co., Inc., Armonk, NY, 1996.

[BYB04] A. Ben-Yishai and D. Burshtein. A discriminative training algorithm for hidden Markov models. *IEEE Trans. On Speech and Audio Processing*, 12(3):204–217, May 2004.

[CB64] R.M. Chapman and H.R. Bragdon. Evoked responses to numerical and nonnumerical visual stimuli while problem solving. *Nature*, 203:1155–1157, 1964.

[CCG91] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, 2:302–309, 1991.

[CG99] J.K. Chapin and G. Gaal. Robotic control from realtime transformation of multi-neuronal population vectors. In *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, page 54, The Rensselaerville Institute, Rensselaerville, New York,, June 1999.

[CMMN99] J.K. Chapin, K.A. Moxon, R.S. Markowitz, and M.A.L. Nicoleslis. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neurosci.*, 2:7:664–670, 1999.

[CS01] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge Univerysity Press, Cambridge, UK, 2000.

[CYB96] E.S. Chng, H. Yang, and S. Bos. Adaptive orthogonal least squares learning algorithm for the radial basis function network. In Usui et al., editor, *IEEE Workshop on Neural network for Signal Processing VI*, pages 3–12, Kyoto 96, 1996.

[DeG70] M.H. DeGroot. *Optimal statistical decisions*. McGraw-Hill, New York, 1970.

[Dev96] S. Devulapalli. Non-linear component analysis and classification of eeg during mental tasks. Master's thesis, Colorado State University, 1996.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.

[dSvLR86] F.H. Lopes da Silva, W.Storm van Leeuwen, and A. Remond. *Handbook of Electroencephalography and Clinical Neurophysiology: Volume 2, Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals*. N. Elsevier Science Publishers, 1986.

[DSW00] E. Donchin, K.M. Spencer, and R. Wijesinghe. The mental prosthesis: assessing the speed of a p300-based brain-computer interface. *IEEE Trans. Rehab. Eng.*, 8:174–179, June 2000.

[FD88] L. A. Farwell and E. Donchin. Talking off the top of your head: Toward a mental prothesis utilizing event-related brain potentials. *Electroenceph. Clin. Neurophysiol.*, 70:510–523, 1988.

[FHR+95] T. Fernandez, T. Harmony, M. Rodriguez, J. Bernal, and J. Silva. Eeg activation patterns during the performance of tasks involving different components of mental calculation. *Electroenceph. Clin. Neurophysiol.*, 94:175–182, 1995.

[For73] G. D. Forney. The viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[Fre91] W.J. Freeman. *Induced rhythms of the brain.* Birkhaeser Boston Inc., 1991.

[Fre95] W.J. Freeman. Chaos in the brain: Possible roles in biological intelligence. *International Journal of Intelligent Systems*, 10:71–88, 1995.

[GG84] Stuart German and Donald German. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intel.*, 6:721–741, 1984.

[GSWP99] C. Guger, A. Schlgl, D. Walterspacher, and G. Pfurtscheller. Design of an eeg-based brain-computer interface (bci) from standard components running in real-time under windows. *Biomed. Technik*, 44:12–16, 1999.

[Hay99] S. Haykin. *Neural Networks. A Comprehensive Foundation.* Prentice Hall, New Jersey, USA, 1999.

[Hjo75] B. Hjorth. On-line transformation of eeg scalp potentials into orthogonal source. *Electroenceph. Clin. Neurophysiol*, 39:111–118, 1975.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning.* Springer-Verlag, New York, 2001.

[Jas58] H.H. Jasper. The ten-twenty electrode system of the international federation. *Electroencephalogram and Clinical Neurophysiology*, 10:371–375, 1958.

[JCL97] B.H Juang, W. Chou, and C.H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. On Speech and Audio Processing*, 5(3):257–265, May 1997.

[JMCM98] K.S. Jones, M.S. Middendorf, G. Calhoun, and G. McMillan. Evaluation of an electroencephalographic-based control device. In *Proc. of the 42nd Annual Mtg of the Human Factors and Ergonomics Society*, pages 491–495, 1998.

[Kan90] K. Kaneko. Globally coupled chaos violates the law of large numbers. *Physical Review Letters*, 65:1391–1394, 1990.

[Kan92] K. Kaneko. Mean field fluctuation in network of chaotic elements. *Physica D*, 55:368–384, 1992.

[KB98] P.R. Kennedy and R.A.E. Bakay. Restoration of neural output from a paralyzed patient by a direct brain connection. *Neuro Report*, 9:1707–1711, 1998.

[KBM+00] P.R. Kennedy, R.A.E. Bakay, M.M. Moore, K. Adams, and J. Goldwaithe. Direct control of a computer from the human central nervous system. *IEEE Trans. Rehab. Eng.*, 8:198–202, June 2000.

[KFN+96] J. Kalcher, D. Flotzinger, Ch. Neuper, S. Golly, and G. Pfurtscheller. Graz brain-computer interface ii: towards communication between humans and computers based on online classification of three different eeg patterns. *Medical and Biological Engineering and Computing*, 34:382–388, 1996.

[Kre99] U. Kre"sel. Pairwise classification and support vector machines. In B. Schölkoph, C.J.C. Burges, and A.J Smola, editors, *Advances in kernel methods: Support Vector Learning*, pages 255–268, 1999.

[LMP01] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 2001.

[LTL93] S. Lin, Y. Tsai, and C Liou. Conscious mental tasks and their eeg signals. *Medical and Biol. Engineering and Comput.*, 31:421–425, 1993.

[MGPF99] J. Müller-Gerking, G. Pfurtcheller, and H. Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, 110:787–798, 1999.

[MMCJ99] M.S. Middendorf, G. McMillan, G. Calhoun, and K.S. Jones. Brain computer interfaces based on the steady-state visual-evoked response. In *Brain-Computer Interface Technology: Theory and Practice: First International Meeting Program and Papers*, pages 78–82, Rensselaerville, New York, June 1999. The Rensselaerville Institute.

[MMCJ00] M. Middendorf, G. McMillan, G. Calhoun, and K.S. Jones. Brain-computer interfaces based on the steady-state visual-evoked response. *IEEE Trans. Rehab. Eng.*, 8:211–214, June 2000.

[MNRW93] D.J. McFarland, G.W. Neat, R.F. Read, and J.R. Wolpaw. An eeg-based method for graded cursor control. *Psychobiology*, 21(1):77–81, 1993.

[MW03] Dennis J. McFarland and Jonathan R. Wolpaw. EEG-based communication and control: Speed-accuracy relationships. *Applied Psychophysiology and Biofeedback*, 28:217–231, September 2003.

[NJ01] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, 2001.

[Nun95] P.L. Nunez. *Neocortical Dynamics and Human EEG Rhythms*. Oxford University Press, New York, 1995.

[OGNP01] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller. Hidden Markov models for online classification of single trial EEG data. *Pattern Recognition Letters*, 22:1299–1309, 2001.

[OMTF96] Marco Onofrj, Donato Melchionda, Astrid Thomas, and Tommaso Fulgente. Reappearance of event-related p3 potential in locked-in syndrome. *Cognitive Brain Research*, 4:95–97, 1996.

[PCSt00] John C. Platt, Nello Cristianini, and John Shawe-taylor. Large margin DAGs for multiclass classification. In *Advances in Neural Information Processing Systems*, volume 12, pages 547–553. MIT Press, 2000.

[PFK93] G. Pfurtscheller, D. Flotzinger, and J. Kalcher. Brain-computer interface - anew communication device for handicapped persons. *J. of Microcomputer Applications*, 16:293–299, 1993.

[PG99] Lopes da Silva FH. Pfurtscheller G. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, 110:1842–1857, November 1999.

[PKN+96] G. Pfurtscheller, J. Kalcher, Ch. Neuper, D. Flotzinger, and M. Pregenzer. On-line eeg classification during externally-paced hand movements using a neural network-based classifier. *Electroenceph. Clin. Neurophysiol.*, 99:416–425, 1996.

[Pla99] John C. Platt. Using analytic qp and sparseness to speed training of support vector machines. In *Advances in Neural Information Processing Systems 11*, pages 557–563, 1999.

[PN01] G. Pfurtscheller and C. Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89:1123–1134, July 2001.

[PNFP97] G. Pfurtscheller, Ch. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and clinical Neurophysiology*, 103:642–651, 1997.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of The IEEE*, 77:257–286, February 1989.

[RBH+00] Jonathan R.Wolpaw, Niels Birbaumer, William J. Heetderks, Dennis J. McFarland, P. Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A. Quatrano, Charles J. Robinson, and Theresa M. Vaughan. Brain computer interface technology: A review of the first international meeting. *IEEE Trans. Rehab. Eng.*, 8:164–173, June 2000.

[SBZJ65] S. Sutton, M. Braren, J. Zublin, and E. John. Evoked potential correlates of stimulus uncertainty. *Science*, 150:1187–1188, 1965.

[Shn98] B. Shneiderman. *Designing the user interface: Strategies for effective humancomputer interaction*. Addison Wesley, Mass., 3rd edition, 1998.

[Spe91] R. Spehlmann. *Spehlmann's EEG Primer*. N. Elsevier Science Publishers, Amsterdam, 1991.

[Sut92] Erich E. Sutter. The brain response interface: communication through visually-induced electrical brain responses. *J. Microcomput. Appl.*, 15(1):31–45, 1992.

[TMU92] N. Toda, N. Murai, and S. Usui. *Artificial Neural Networks 2*, chapter A measure of nonlinearity in time series using neural network prediction model, pages 1117–1120. 1992.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[Vau03] T.M Vaughan. Guest editorial brain-computer interface technology: a review of the second international meeting. *IEEE Trans. Rehab. Eng.*, 11:94–109, June 2003.

[Vid73] Jacques J. Vidal. Toward direct brain-computer communication. In L. J. Mullins, editor, *Annual Review of Biophysics and Bioengineering*, pages 157–180. Annual Reviews Inc., 1973.

[Vit67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, April 1967.

[VWD96] T.M. Vaughan, J.R. Wolpaw, and E. Donchin. Eeg-based communication: Prospects and problems. *IEEE Trans. on Rehabilitation Engineering*, 4(4):425–430, 1996.

[WBM+02] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarl, Gert Pfurtscheller, and Theresa M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, June 2002.

[WL96] J.J. Wright and D.T.J. Liley. Dynamics of the brain at global and microscopic scales: Neural networks and the eeg. *Behavioral and Brain Sciences*, 19:285–320, 1996.

[WMNF91] J.R. Wolpaw, D.J. McFarland, G.W. Neat, and C.A. Forneris. An eeg-based brain-computer interface for cursor control. *Electroenceph. Clin. Neurophysiol.*, 78:252–258, 1991.

[WW99] J. Weston and C. Watkins. Multi-class support vector machines. In M. Verleysen, editor, *Proceeding of ESANN99*, Brussels, 1999. D. Facto Press.