CHAPTER 1

Probabilistic Approaches to ECG Segmentation and Feature Extraction

Nicholas P. Hughes

11.1 Introduction

The development of new drugs by the pharmaceutical industry is a costly and lengthy process, with the time from concept to final product typically lasting 10 years. Perhaps the most critical stage of this process is the phase one study, where the drug is administered to humans for the first time. During this stage each subject is carefully monitored for any unexpected adverse effects which may be brought about by the drug. Of particular interest is the ECG of the patient, which provides detailed information about the state of the patient’s heart.

By examining the ECG signal in detail, it is possible to derive a number of informative measurements from the characteristic ECG waveform. These can then be used to assess the medical well-being of the patient, and more importantly, detect any potential side effects of the drug on the cardiac rhythm. The most important of these measurements is the QT interval. In particular, drug-induced prolongation of the QT interval (so called Long QT Syndrome) can result in a very fast, abnormal heart rhythm known as torsade de pointes. This rhythm can degenerate into ventricular fibrillation and hence lead to sudden cardiac death.

In practice, QT interval measurements are carried out manually by specially trained ECG analysts. This is an expensive and time-consuming process, which is susceptible to mistakes by the analysts and provides no associated degree of confidence (or accuracy) in the measurements. This problem was recently highlighted in the case of the antihistamine terfenadine, which had the side effect of significantly prolonging the QT interval in a number of patients. Unfortunately this side effect was not detected in the clinical trials and only came to light after a large number of people had unexpectedly died while taking the drug [1].

In this chapter we consider the problem of automated ECG interval analysis from a probabilistic modeling perspective. In particular, we examine the use of hidden Markov models for automatically segmenting an ECG signal into its constituent waveform features. An undecimated wavelet transform is used to provide an informative representation which is both robust to noise and tuned to the morphological characteristics of the waveform features. Finally we investigate the use of duration constraints for improving the robustness of the model segmentations.
11.2 The Electrocardiogram

11.2.1 The ECG Waveform

Each individual heartbeat is comprised of a number of distinct cardiological stages, which in turn give rise to a set of distinct features in the ECG waveform. These features represent either *depolarization* (electrical discharging) or *repolarization* (electrical recharging) of the muscle cells in particular regions of the heart. Figure 11.1 shows a human ECG waveform and the associated features. The standard features of the ECG waveform are the P wave, the QRS complex, and the T wave. Additionally a small U wave (following the T wave) is occasionally present.

The cardiac cycle begins with the P wave (the start and end points of which are referred to as $P_{on}$ and $P_{off}$), which corresponds to the period of *atrial depolarization* in the heart. This is followed by the QRS complex, which is generally the most recognizable feature of an ECG waveform, and corresponds to the period of *ventricular depolarization*. The start and end points of the QRS complex are referred to as the Q and J points. The T wave follows the QRS complex and corresponds to the period of *ventricular repolarization*. The end point of the T wave is referred to as $T_{off}$ and represents the end of the cardiac cycle (presuming the absence of a U wave).

11.2.2 ECG Interval Analysis

The timing between the onset and offset of particular features of the ECG (referred to as an *interval*) is of great importance since it provides a measure of the state of the heart and can indicate the presence of certain cardiological conditions. Two of

![Figure 11.1](image.jpg)  
*Figure 11.1* A typical human ECG waveform and its associated feature boundaries.
the most important intervals in the ECG waveform are the QT interval and the PR interval. The QT interval is defined as the time from the start of the QRS complex to the end of the T wave (i.e., $T_{off} - Q$) and corresponds to the total duration of electrical activity (both depolarization and repolarization) in the ventricles. Similarly, the PR interval is defined as the time from the start of the P wave to the start of the QRS complex (i.e., $Q - P_{on}$) and corresponds to the time from the onset of atrial depolarization to the onset of ventricular depolarization.

Changes in the QT interval are currently the gold standard for evaluating the effects of drugs on ventricular repolarization. In addition, changes in the PR interval can indicate the presence of specific cardiological conditions such as atrioventricular block [2]. Thus, the accurate measurement and assessment of the QT and PR intervals is of paramount importance in clinical drug trials.

11.2.3 Manual ECG Interval Analysis

Manual ECG interval analysis is typically performed by specialist ECG analysis companies known as centralized ECG core laboratories. The expert analysts (or “readers”) employed by these labs are generally a mixture of professional cardiologists and highly trained cardiac technicians.

The accurate measurement of the QT interval is made difficult by the need to locate the end of the T wave to a high level of precision. In theory, the end of the T wave is defined as the point at which the ECG signal (for the T wave) returns to the isoelectric baseline. In practice, however, determining this point precisely is challenging due to the variation in the baseline amplitude, unusual or abnormal T wave morphologies (such as T-U fusions or flat T waves), and the presence of noise or artifact in the signal.

As a result, T wave offset measurements by expert analysts are inherently subjective and the associated QT interval measurements often suffer from a high degree of interanalyst and intra-analyst variability. There has therefore been much focus on the problem of developing an automated ECG interval analysis system, which could provide robust and consistent measurements, together with an associated degree of confidence in each measurement [3].

11.3 Automated ECG Interval Analysis

Standard approaches to ECG segmentation attempt to find the ECG waveform feature boundaries from a given ECG signal in a number of successive stages [4]. In the first stage, a standard QRS detection algorithm (such as the Pan and Tompkins algorithm [5]) is used to locate the R peaks in the ECG signal. Given the location of the R peak in each ECG beat, the next stage in the process is to search forwards and backwards from this point to estimate the locations of the onset and offset boundaries for the various ECG features [4].

In common with manual analysis, the accurate determination of the end of the T wave with automated methods is a challenging problem. The standard approach to this problem is based on the tangent method, which is illustrated in Figure 11.2. This technique, which was first introduced in 1952, locates the end of the T wave as the
point of intersection between the (estimated) isoelectric baseline and the tangent to the downslope of the T wave (or the upslope for inverted T waves) [6]. The tangent itself is computed at the point of maximum (absolute) gradient following the peak of the T wave.

A significant disadvantage of the tangent method is that it is sensitive to the amplitude of the T wave. In particular, large amplitude T waves can cause the tangent to intersect the isoelectric baseline well before the actual end of the T wave. As a result, the automated QT intervals from the tangent method can significantly underestimate the actual QT interval value. To overcome this problem, Xu and Reddy applied a “nonlinear correction” factor (based on the T wave amplitude) to the T wave offset determined by the tangent method [7].

In practice, automated QT interval measurements continue to be unreliable in the presence of noisy waveforms or unusual T wave morphologies. In addition, it is still the case that automated systems can produce highly unreliable measurements for relatively clean ECGs with regular waveform morphologies [8]. As a result, ECG interval measurements for clinical drug trials are generally performed manually by human experts.

11.4 The Probabilistic Modeling Approach

The probabilistic modeling approach to automated ECG interval analysis offers a number of significant advantages compared with traditional approaches. In particular, probabilistic models provide the following benefits:
11.4 The Probabilistic Modeling Approach

- The ability to learn from ECG data sets annotated by human experts;
- The ability to incorporate prior knowledge about the statistical characteristics of the ECG waveform features in a principled manner;
- The ability to produce confidence measures in the automated ECG interval measurements.

One of the most powerful features of the probabilistic modeling approach is the ability to learn a model for ECG segmentation from expert annotated ECG data. Such data-driven learning allows us to sidestep neatly the problem of having to specify an explicit rule for determining the end of the T wave in the electrocardiogram (e.g., the tangent method). Since rule-based approaches are inherently unreliable for this particular problem (given the wide range of ECG morphologies which can occur in practice), it is much more appropriate to learn the statistical characteristics which define the ECG waveform feature boundaries using measurements from expert analysts. Furthermore, this learning approach enables us to build models from ECG data sets which exhibit specific types of waveform morphologies, or from those corresponding to specific patient cohorts [9].

The most appropriate form of probabilistic model to apply to the task of automated ECG interval analysis is a hidden Markov model (HMM). These models can be viewed as a form of state space model. Specifically, HMMs make use of a discrete state space to capture the sequential characteristics of the data. This particular formulation offers an attractive framework for ECG signal modeling, since it allows each of the ECG waveform features (i.e., P wave, QRS complex, T wave, and so forth) to be uniquely associated with a particular state in the model. Thus, the model is able to take advantage of the sequence ordering which exists between the different waveform features of normal ECGs.

An additional advantage of utilizing hidden Markov models when considering probabilistic approaches to ECG segmentation is that there are efficient algorithms for HMM training and testing. Specifically, the parameters of an HMM can be estimated straightforwardly in a supervised manner using a data set of ECG waveforms together with the corresponding expert measurements of the waveform feature boundaries. Alternatively, if the data set contains only the ECG waveforms (i.e., the associated expert measurements are unavailable), then the EM algorithm can be used to train the model in an unsupervised manner [10]. Once the model has been trained, the Viterbi algorithm can be used to segment test ECG signals.

A disadvantage of the probabilistic modeling approach is that the segmentation of ECGs typically requires more computing power (for the same level of processing performance) compared with standard algorithms. For the analysis of 10-second 12-lead ECG signals, this is rarely an issue. However, when very large amounts of ECG data must be processed rapidly (e.g., for the analysis of continuous 12-lead digital Holter recordings), it may be necessary to combine both standard (nonprobabilistic) and probabilistic algorithms to achieve an appropriate rate of segmentation.

Although the use of hidden Markov models for ECG analysis has been considered previously in the literature [11, 12], many of the issues involved in achieving a high level of segmentation accuracy with these models have not previously been discussed in detail. In this chapter, we focus on the core issues which must be addressed...
when using probabilistic models for automated ECG interval analysis. Specifically, we consider the choice of representation for the ECG signal, and the choice of model architecture for the segmentation.

In the following sections, we describe how wavelet methods (and in particular, the undecimated wavelet transform) can be used to provide an encoding of the ECG which is more appropriate for subsequent modeling with HMMs. In addition, we describe how “duration constraints” can be incorporated into the HMM framework such that the resulting model provides a more appropriate statistical description of the normal ECG waveform, and hence a greater degree of robustness in the waveform segmentations.

11.5 Data Collection

In order to develop an automated system for ECG interval analysis, we collected a data set of over 100 ECG waveforms (sampled at 500 Hz), together with the corresponding waveform feature boundaries as determined by a group of expert ECG analysts. Due to time constraints it was not possible for each expert analyst to label every ECG waveform in the data set. Therefore, we chose to distribute the waveforms at random among the different experts (such that each waveform was measured by one expert only).

For each ECG waveform, the following points were annotated: P\text{on}, P\text{off}, Q, J, and T\text{off} (if a U wave was present, the U\text{off} point was also annotated). In addition, the point corresponding to the start of the next P wave (i.e., the P wave of the following heartbeat), NP\text{on}, was also annotated. During the data collection exercise, we found that it was not possible to obtain reliable estimates for the Ton and Uon points, and therefore these were taken to be the J and T\text{off} points, respectively.

11.6 Introduction to Hidden Markov Modeling

11.6.1 Overview

Since their development in the late 1960s and early 1970s, hidden Markov models have proved to be a powerful and flexible class of statistical model for describing many different kinds of sequential data. The term “sequential” here refers to the fact that there exists a natural ordering inherent in the data itself. This property is particularly true of time-series data, where the individual data samples are ordered according to the particular time point at which they were measured. By incorporating this sequence information into the structure of our model, we can ensure that the model provides a good description of the data and its associated statistical properties.

The utility of hidden Markov models stems from the fact that they offer an effective balance between the core data modeling issues of complexity and tractability. In particular, hidden Markov models are “rich” enough to provide a good statistical description of many different kinds of sequence data, yet they are also sufficiently simple as to admit efficient algorithms for inference and learning. This trade-off between descriptive modeling power and practical ease of use is perhaps the main reason for the success of hidden Markov models in practice.
This section presents a thorough review of hidden Markov models and their associated algorithms for inference and learning. We begin with a brief description of Markov models in the context of stochastic processes, and then proceed to cover the topic of hidden Markov models in more depth.

11.6.2 Stochastic Processes and Markov Models

A natural way to describe a stochastic process is in terms of the probability distribution of the random variable under consideration. In particular, for processes that evolve through time, it is often useful to consider a conditional probability distribution of the form

\[ p(x_t | x_{t_1}, x_{t_2}, x_{t_3}, \ldots) \]  

which defines the probability of obtaining a sample value \( x \) at time \( t \) given a history of previous values. If this distribution is time-invariant, such that it is only dependent on the time differences (or “lags”) and not the absolute time values, then the process is said to be strictly stationary.\(^1\) In this case, (11.1) can be written in the form:

\[ p(x_t | x_{t-\tau}, x_{t-\tau_2}, x_{t-\tau_3}, \ldots) \]  

A special case of interest occurs when this conditional distribution is dependent on only a finite history of previous values, such that:

\[ p(x_t | x_{t-\tau_1}, x_{t-\tau_2}, \ldots, x_{t-\tau_N}) \]  

which defines an Nth-order Markov process. If we now make the further simplifying assumption that the process depends solely on the previous value, then the conditional distribution becomes

\[ p(x_t | x_{t-\tau}) \]  

This equation defines a first-order Markov process, which is often referred to simply as a “Markov process” [13]. When dealing with a discrete-time discrete-valued random variable \( s_t \), this Markov process becomes a Markov chain, with the corresponding conditional distribution:

\[ P(s_t | s_{t-1}) \]  

A Markov model can be used to represent any random variable \( s_t \) which can occupy one of \( K \) possible discrete “states” at each time step and which satisfies the Markov property

\[ P(s_{t+1} | s_t, s_{t-1}, s_{t-2}, \ldots) = P(s_{t+1} | s_t) \]  

Equation (11.6) captures the notion that “the future is conditionally independent of the past given the present.” Thus when evaluating the probability of the

---

1. A more relaxed definition of stationarity requires that only the first and second moments of the distribution are time-invariant, in which case the process is said to be weakly stationary.
system state at a particular time step, we need only consider the state at the previous time step. Statistical models which exploit this Markov property often admit efficient algorithms for computing many practical quantities of interest.

In practice, a Markov model is governed by two distinct parameters: an initial state distribution and a state transition matrix. The initial state distribution \( \pi \) defines the probabilities of the random variable being in each of the \( K \) possible states at the first time step (i.e., \( t = 1 \)). Thus, this parameter is simply a \( K \)-dimensional vector, where each element \( \pi_k \) gives the corresponding probability \( P(s_1 = k) \). The state transition matrix \( A \) defines the probability of the model “transitioning” to a particular state at the next time step, given its state at the current time step. Thus, this parameter is a \( K \times K \) matrix, where each element \( a_{ij} \) gives the corresponding probability \( P(s_{t+1} = j \mid s_t = i) \). If it is possible to transition from any state to any other state (i.e., \( a_{ij} \neq 0 \ \forall i, j \)), then the model is said to be ergodic [7].

### 11.6.3 Hidden Markov Models

A hidden Markov model is a probabilistic model which describes the statistical relationship between an observable sequence \( O \) and an unobservable or “hidden” state sequence \( S \). The hidden state itself is discrete and governed by an underlying Markov model. The observation values however may be either continuous or discrete in nature.

The key aspect of an HMM is that each observation value is considered to be the result of an additional stochastic process associated with one of the hidden states. Thus, a hidden Markov model can be viewed as a “doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations” [10].

More formally, an HMM (with \( K \) hidden states) is defined by the following three parameters [10]:

- An initial state distribution \( \pi \);
- A state transition matrix \( A \);
- An observation probability distribution \( b_k \) for each state \( k \).

The first two parameters govern the underlying Markov model which describes the statistical properties of the hidden states. It is the observation probability distributions\(^2\) however which differentiate a hidden Markov model from a standard Markov model. More precisely, the observation probability distribution for a given state models the probability of a particular observation value when the model occupies that particular state.

It is often useful to consider a hidden Markov model from a generative perspective. That is, we can consider the HMM as providing a bottom-up description of how the observed sequence \( O \) is produced or generated. Viewed as a generative model, the operation of an HMM is as follows:

---

2. The observation probability distributions are also known as emission probability distributions.
11.6 Introduction to Hidden Markov Modeling

1. Select the initial state \( k \) by sampling from the initial state distribution \( \pi \).
2. Generate a observation value from this state by sampling from the associated observation distribution \( b_k \).
3. Select the state at the next time step based upon the transition matrix \( A \).
4. Return to step 2.

In the standard formulation of a hidden Markov model, the observation values “within” a given state are considered to be independent and identically distributed (i.i.d.). Hence, when an observation value is generated by a particular state at a given time step, it is generated independently of any previous samples which may have been generated from that same state in previous time steps. Thus, conditioned on the state sequence, we can express the likelihood of a sequence of observations as

\[
p(O_{1:T} | S_{1:T}, \lambda) = \prod_{t=1}^{T} p(O_t | S_t, \lambda)
\]

where \( \lambda \) represents the set of model parameters for the HMM. It is important to recognize that the factorization shown in (11.7) holds only when the observations are conditioned on the state sequence. Thus, without knowledge of the underlying state sequence, we cannot make any independence assumptions about the probability distribution of the observations [i.e., \( p(O_{1:T} | \lambda) \)].

The assumption of statistical independence between successive observations (within a state) is perhaps the greatest weakness of the standard hidden Markov model. The validity of this assumption in the context of ECG signal modeling is considered in greater detail in Section 11.7.4.

Figure 11.3 shows two different graphical representations of a simple two-state hidden Markov model. The first representation, illustrated in Figure 11.3(a), shows the “architectural” view of an HMM. This form of the model highlights the overall HMM topology together with the role of the individual model parameters. In particular, the two clear nodes correspond to the two individual hidden states.
in the model, and the two shaded nodes correspond to the observations that are generated by those hidden states.

The second representation, illustrated in Figure 11.3(b), shows the HMM as a dynamic Bayesian network (DBN) [14]. Here the HMM is shown “unrolled” over a number of time steps. Each clear node now corresponds to the particular state which the model occupies at a given time step, and each shaded node corresponds to the associated observation generated by that state. Both graphical representations serve to illustrate the role of an HMM as a statistical model for sequential data.

Hidden Markov models can also be seen as generalizations of various statistical models [15, 16]. Figure 11.4 shows two such possible “views” of an HMM. The first, shown in Figure 11.4(a), denotes the HMM as a simple Markov model with stochastic observations. This is the view outlined previously in this section. An alternative perspective, however, can be gained by considering the HMM as a form of temporal mixture model, as illustrated in Figure 11.4(b). With a standard (static) mixture model, each data point is considered to have been “generated” by one of the $K$ mixture components independently of the other data points [17]. If we now relax the strong assumption of statistical independence between data points and allow the individual mixture components to possess Markovian “dynamics,” the result is a hidden Markov model (where the hidden state at each time step corresponds to the particular mixture component which is active at that time step).

More generally, hidden Markov models can be viewed under the general framework of probabilistic graphical models [18]. Such models include standard “state-space” models such as hidden Markov models and Kalman filters as special cases, as well as more advanced models such as coupled HMMs [19] and factorial HMMs [20].

From a purely statistical perspective, an HMM defines a joint probability distribution over observation sequences $O$ and hidden state sequences $S$ [i.e., $p(O, S | \lambda)$]. Given this joint distribution, it is often of interest to find the particular state sequence which maximizes the conditional distribution $P(S | O, \lambda)$. This corresponds to

---

3. In the “language” of Bayesian networks, a square node represents a discrete value and a circular node represents a continuous value. Similarly a clear node represents a hidden (or latent) variable and a shaded node represents an observed variable. Hence in Figure 11.3(b), the hidden states are shown as clear square nodes, and the observations (which are assumed to be continuous in this example) are shown as shaded circular nodes.
the state sequence which is most likely to have “generated” the given observation sequence, and thus provides an effective means to segment the observation sequence into its characteristic features. However, before we can use an HMM for such purposes, we must first learn the “optimal” parameters of the model from a given data set (such that the HMM provides a useful statistical model of the data). We now consider the solution to each of these problems in greater detail.

11.6.4 Inference in HMMs

The inference problem for hidden Markov models is typically cast as the problem of determining the single most probable state sequence given the observation data, that is,

\[ S^* = \arg \max_S \{ P(S \mid O, \lambda) \} \] (11.8)

This can be reexpressed using Bayes’ rule as

\[ S^* = \arg \max_S \{ P(S \mid O, \lambda) \} = \arg \max_S \{ P(S, O \mid \lambda) \} = \arg \max_S \{ p(S, O \mid \lambda) \} \] (11.9)

Hence it suffices to find the state sequence \( S^* \) which maximizes the joint distribution \( p(S, O \mid \lambda) \). The solution to this problem is known as the Viterbi algorithm [21, 22]. More precisely, the Viterbi algorithm is a dynamic programming procedure which takes advantage of the Markov property of the HMM state sequence.

To apply dynamic programming to the HMM inference problem, we must first define the variable \( \delta_t(i) \):

\[ \delta_t(i) = \max_{s_1s_2 \cdots s_{t-1}} \{ p(s_1s_2 \cdots s_t = i, O_1O_2 \cdots O_t \mid \lambda) \} \] (11.10)

This is the likelihood of the most probable state sequence that accounts for the first \( t \) observations and ends in state \( i \) at time \( t \). Now consider computing the value of the delta variable at the next time step \( t+1 \). We can express this computation as

\[ \delta_{t+1}(i) = \max_{s_1s_2 \cdots s_t \cdots s_{t+1}} \{ p(s_1s_2 \cdots s_{t+1} = i, O_1O_2 \cdots O_{t+1} \mid \lambda) \} \]

\[ = \max_j \{ \max_{s_1s_2 \cdots s_{t-1}} \{ p(s_1s_2 \cdots s_t = j, O_1O_2 \cdots O_t \mid \lambda) \} \times p(s_{t+1} = i \mid s_t = j) \} p(O_{t+1} \mid s_{t+1} = i) \]

\[ = \max_j \{ \delta_t(j) a_{ji} \} b_i(O_{t+1}) \] (11.11)

The key step in developing the recurrence relation for \( \delta_{t+1}(i) \) is to note that we can make use of the solutions to the previous subproblems at time step \( t \).
In particular, we can compute the most probable state sequence that accounts for the first \( t + 1 \) observations and ends in state \( i \) by maximizing over the \( K \) previous solutions [i.e., \( \delta_l(j) \)] and the appropriate transition probability \( (a_{ji}) \).

The recursion is initialized for each state \( i \) by computing the probability of the model occupying state \( i \) at the first time step and producing the first observation value \( O_1 \) from that particular state, that is,

\[
\delta_1(i) = \pi_i \cdot b_i(O_1)
\] (11.12)

Equation (11.11) can then be used to compute the value of \( \delta(t) \) for each state \( i \) and for each time step from \( t = 2 \) to \( t = T \). Following the final computation at \( t = T \), we have

\[
\delta_T(i) = \max_{s_1 s_2 \cdots s_{T-1}} \{ p(s_1 s_2 \cdots s_T = i, O_1 O_2 \cdots O_T | \lambda) \} \] (11.13)

The optimal value for the hidden state at the final time step is then computed as the particular state which maximizes (11.13)

\[
s_T^* = \argmax_i \{ \delta_T(i) \} \] (11.14)

Using this knowledge of the optimal state value at the final time step, we can then “work back” to uncover the optimal state value at the previous time step \( t = T - 1 \). This is given by the particular state argument which maximized \( \delta_T(s_T^*) \) as part of its recursive computation. Based on this value, we can follow a similar procedure to uncover the optimal state value at time step \( T - 2 \). This general backtracking procedure can be performed successively to uncover the full optimal hidden state sequence \( S^* \).

When computing (11.11) in practice, it is common to record the particular “maximizing” state which maximizes the value of \( \delta_{t+1}(i) \):

\[
\psi_{t+1}(i) = \argmax_j \{ \delta_t(j) a_{ji} \} \] (11.15)

The back-tracking procedure to uncover the optimal state sequence can then be implemented as a look-up process using the stored \( \psi \) values:

\[
s_t^* = \psi_{t+1}(s_{t+1}^*) \quad t = T - 1, T - 2, \cdots, 1 \] (11.16)

Pseudo-code for the Viterbi procedure is shown in Listing 1. Note that in practice, the implementation of the Viterbi algorithm requires the use of a simple “scaling” procedure to ensure that the \( \delta_t(i) \) values do not under- or overflow [10].

11.6.5 Learning in HMMs

The learning problem in hidden Markov models is concerned with determining the optimal model parameters given a particular training data set. If the data set consists of both the observation sequences and the corresponding hidden state sequences (which generated the observation sequences), then the HMM parameter
11.6 Introduction to Hidden Markov Modeling

Listing 1 Viterbi algorithm

//Initialization:
for \( i = 1 \) to \( K \)
\[ \delta_1(i) = \pi_i b_i(O_1) \]
\[ \psi_1(i) = 0 \]
end

//Recursion:
for \( t = 2 \) to \( T \)
//Compute delta at time \( t \) for each state \( i \)
//and record “maximizing” predecessor state
for \( i = 1 \) to \( K \)
\[ \delta_t(i) = \max_{1 \leq j \leq K} \{ \delta_{t-1}(j) a_{ji} b_i(O_t) \} \]
\[ \psi_t(i) = \arg\max_{1 \leq j \leq K} \{ \delta_{t-1}(j) a_{ji} \} \]
end
end

//Termination:
\[ P^* = \max_{1 \leq i \leq K} \{ \delta_T(i) \} \]
\[ s^*_T = \arg\max_{1 \leq i \leq K} \{ \delta_T(i) \} \]
//Backtracking:
for \( t = T - 1 \) to \( 1 \)
\[ s^*_t = \psi_{t+1}(s^*_{t+1}) \]
end

estimation problem can be viewed as a supervised learning problem. Conversely, if the data set consists of only the observation sequences, then the problem is one of unsupervised learning. We now consider each of these two cases in turn.

11.6.5.1 Supervised Learning

In the supervised learning case, we can make use of both the observation sequences and the corresponding hidden state sequences to derive simple estimates for the model parameters. In particular, we can estimate the initial state distribution by evaluating the fraction of the hidden state sequences which commence in each of the given model states at the first time step. More precisely, denoting the total number of hidden state sequences which commence in state \( i \) at the first time step by \( n_{\text{init}}(i) \), then we have the following estimator for the \( i \)th element of the initial state distribution:

\[ \pi_i = \frac{n_{\text{init}}(i)}{\sum_{k=1}^{K} n_{\text{init}}(k)} \quad (11.17) \]

In a similar manner, we can estimate the transition matrix by evaluating the fraction of particular state transitions over all the hidden state sequences. More precisely, denoting the total number of transitions from state \( i \) to state \( j \) over all the hidden state sequences by \( n_{\text{trans}}(i, j) \), then we have the following estimator for the \((i, j)\)th element of the transition matrix:

\[ a_{ij} = \frac{n_{\text{trans}}(i, j)}{\sum_{k=1}^{K} n_{\text{trans}}(i, k)} \quad (11.18) \]
The exact estimator for the parameters of the observation models depends on the specific functional form chosen for these models. However, the general estimation procedure for the observation models is straightforward. For each state $i$, we simply “extract” all the observations which correspond to that particular state (i.e., those that were “generated” by that state) and then fit the observation model to this data in a standard manner.

### 11.6.5.2 Unsupervised Learning

In the *unsupervised* learning case, we are provided with a data set of observation sequences only (i.e., we do not have access to the corresponding hidden state sequences). This makes the learning procedure much more difficult compared with the supervised case previously described.

Given the data set of observation sequences $O = \{O_1, O_2, \ldots, O_N\}$, the unsupervised learning problem is typically cast in a maximum likelihood framework. More precisely, we seek the model parameters $\lambda^*$ which maximize the probability of the data; that is,

$$\lambda^* = \arg\max_{\lambda} \{ p(O | \lambda) \} \tag{11.19}$$

Fortunately, there exists a particularly effective approach to solving (11.19) for hidden Markov models (and many other types of statistical models). This approach is known as the EM algorithm, and is a general method for unsupervised learning in the presence of “missing” or incomplete data [23].

We now discuss the application of hidden Markov models to the particular problem of ECG segmentation.

### 11.7 Hidden Markov Models for ECG Segmentation

This section presents a detailed analysis of the use of hidden Markov models for *segmenting* ECG signals. The general aim of any signal segmentation method is to “partition” a given signal into consecutive regions of interest. In the context of the ECG then, the role of segmentation is to determine as accurately as possible the onset and offset boundaries of the various waveform features (e.g., P wave, QRS complex, T wave, and so forth), such that the ECG interval measurements may be computed automatically.

In Section 11.7.1, we discuss the different types of hidden Markov model architecture which can be used for ECG segmentation. Following this, we discuss the two different forms of segmentations which can occur when a trained HMM is used to segment ECG signals in practice. The performance of HMMs for ECG segmentation is then considered in more detail. In particular, we examine a number of different state observation models, as well as the use of ECG signal normalization techniques.

---

4. In the context of hidden Markov models, the EM algorithm is often referred to as the *Baum-Welch algorithm*. 

11.7 Hidden Markov Models for ECG Segmentation

11.7.1 Overview

The first step in applying hidden Markov models to the task of ECG segmentation is to associate each state in the model with a particular region of the ECG. As discussed previously in Section 11.6.5, this can either be achieved in a supervised manner (i.e., using expert measurements) or an unsupervised manner (i.e., using the EM algorithm). Although the former approach requires each ECG waveform in the training data set to be associated with expert measurements of the waveform feature boundaries (i.e., the $P_{on}$, $Q$, $T_{off}$ points, and so forth), the resulting models generally produce more accurate segmentation results compared with their unsupervised counterparts.

Figure 11.5 shows a variety of different HMM architectures for ECG interval analysis. A simple way of associating each HMM state with a region of the ECG is to use individual hidden states to represent the P wave, QRS complex, JT interval and baseline regions of the ECG, as shown in Figure 11.5(a). In practice, it is advantageous to partition the single baseline state into multiple baseline states, one of which is used to model the baseline region between the end of the P wave and the start of the QRS complex (termed “baseline 1”), and another which is used to model the baseline region following the end of the T wave (termed “baseline 2”). This model architecture, which is shown in Figure 11.5(b), will be used throughout the rest of this chapter.5

Following the choice of model architecture, the next step in training an HMM is to decide upon the specific type of observation model which will be used to capture the statistical characteristics of the signal samples from each hidden state. Common choices for the observation models in an HMM are the Gaussian density, the Gaussian mixture model (GMM), and the autoregressive (AR) model. Section 11.7.4 discusses the different types of observation models in the context of ECG segmentation.

Before training a hidden Markov model for ECG segmentation, it is beneficial to consider the use of preprocessing techniques for ECG signal normalization.

11.7.2 ECG Signal Normalization

In many pattern recognition tasks it is advantageous to normalize the raw input data prior to any subsequent modeling [24]. A particularly simple and effective form of signal normalization is a linear rescaling of the signal sample values. In the case of the ECG, this procedure can help to normalize the dynamic range of the signal and to stabilize the baseline sections.

A useful form of signal normalization is given by range normalization, which linearly scales the signal samples such that the maximum sample value is set to +1 and the minimum sample value to −1. This can be achieved in a simple two-step process. First, the signal samples are “amplitude shifted” such that the minimum and maximum sample values are equidistant from zero. Next, the signal samples are linearly scaled by dividing by the new maximum sample value. These two steps

---

5. Note that it is also possible to use an “optional” U wave state (following the T wave) to model any U waves that may be present in the data, as shown in Figure 11.5(c).
can be stated mathematically as

\[ x'_n = x_n - \left( \frac{x_{\min} + x_{\max}}{2} \right) \]  

(11.20)

and

\[ y_n = \frac{x'_n}{x'_{\max}} \]  

(11.21)
11.7 Hidden Markov Models for ECG Segmentation

where $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum values in the original signal, respectively. The range normalization procedure can be made more robust to the presence of artefact or “spikes” in the ECG signal by computing the median of the minimum and maximum signal values over a number of different signal segments. Specifically, the ECG signal is divided evenly into a number of contiguous segments, and the minimum and maximum signal values within each segment are computed. The ECG signal is then range normalized (i.e., scaled) to the median of the minimum and maximum values over the given segments.

11.7.3 Types of Model Segmentations

Before considering in detail the results for HMMs applied to the task of ECG segmentation, it is advantageous to consider first the different types of ECG segmentations that can occur in practice. In particular, we can identify two distinct forms of model segmentations when a trained HMM is used to segment a given 10-second ECG signal:

- **Single-beat segmentations**: Here the model correctly infers only one heartbeat where there is only one beat present in a particular region of the ECG signal.
- **Double-beat segmentations**: Here the model incorrectly infers two or more heartbeats where there is only one beat present in a particular region of the ECG signal.

Figure 11.6(a, b) shows examples of single-beat and double-beat segmentations, respectively. In the example of the double-beat segmentation, the model incorrectly infers two separate beats in the ECG signal shown. The first beat correctly locates the QRS complex but incorrectly locates the end of the T wave (in the region of baseline prior to the T wave). The second beat then “locates” another QRS complex (of duration one sample) around the onset of the T wave, but correctly locates the end of the T wave in the ECG signal. The specific reason for the occurrence of double-beat segmentations and a method to alleviate this problem are covered in Section 11.9.

In the case of a single-beat segmentation, the segmentation errors can be evaluated by simply computing the discrepancy between each individual automated annotation (e.g., $T_{\text{off}}$) and the corresponding expert analyst annotation. In the case of a double-beat segmentation, however, it is not possible to associate uniquely each expert annotation with a corresponding automated annotation. Given this, it is therefore not meaningful to attempt to evaluate a measure of annotation “error” for double-beat segmentations. Thus, a more informative approach is simply to report the percentage of single-beat segmentations for a given ECG data set, along with the segmentation errors for the single-beat segmentations only.

11.7.4 Performance Evaluation

The technique of cross-validation [24] was used to evaluate the performance of a hidden Markov model for automated ECG segmentation. In particular, five-fold cross-validation was used. In the first stage, the data set of annotated ECG
waveforms was partitioned into five subsets of approximately equal size (in terms of the number of annotated ECG waveforms within each subset). For each “fold” of the cross-validation procedure, a model was trained in a supervised manner using all the annotated ECG waveforms from four of the five subsets. The trained model was then tested on the data from the remaining subset. This procedure was repeated for each of the five possible test subsets. Prior to performing cross-validation, the complete data set of annotated ECG waveforms was randomly permuted in order to remove any possible ordering which could affect the results.

As previously stated, for each fold of cross-validation a model was trained in a supervised manner. The transition matrix was estimated from the training waveform annotations using the supervised estimator given in (11.18). For Gaussian observation models, the mean and variance of the full set of signal samples were computed for each model state. For Gaussian mixture models, a combined MDL
and EM algorithm was used to compute the optimal number of mixture components and the associated parameter values [25]. For autoregressive or AR models, the Burg algorithm [26] was used to infer the model parameters and the optimal model order was computed using an MDL criterion.

Following the model training for each fold of cross-validation, the trained HMM was then used to segment each 10-second ECG signal in the test set. The segmentation was performed by using the Viterbi algorithm to infer the most probable underlying sequence of hidden states for the given signal. Note that the full 10-second ECG signal was processed, as opposed to just the manually annotated ECG beat, in order to more closely match the way an automated system would be used for ECG interval analysis in practice.

Next, for each ECG, the model annotations corresponding to the particular beat which had been manually annotated were then extracted. In the case of a single-beat segmentation, the absolute differences between the model annotations and the associated expert analyst annotations were computed. In the case of a double-beat segmentation, no annotation errors were computed. Once the cross-validation procedure was complete, the five sets of annotation “errors” were then averaged to produce the final results.

Table 11.1 shows the cross-validation results for HMMs trained on the raw ECG signal data. In particular, the table shows the percentage of single-beat segmentations and the annotation errors for different types of HMM observation models and with/without range normalization, for ECG leads II and V2.

The results for each lead demonstrate the utility of normalizing the ECG signals (prior to training and testing) with the range normalization method. In each case, the percentage of single-beat segmentations produced by an HMM (with a Gaussian observation model) is considerably increased when range normalization is employed. For lead V2, it is notable that the annotation errors (evaluated on the single-beat segmentations only) for the model with range normalization are greater than those for the model with no normalization. This is most likely to be due to the fact that the latter model produces double-beat segmentations for those waveforms that naturally give rise to larger annotation errors (and hence these waveforms are excluded from the annotation error computations for this model).

The most important aspect of the results is the considerable performance improvement gained by using autoregressive observation models as opposed to Gaussian or Gaussian mixture models. The use of AR observation models enables each HMM state to capture the statistical dependencies between successive groups of observations. In the case of the ECG, this allows the HMM to take account of the shape of each of the ECG waveform features. Thus, as expected, these models lead to a significant performance improvement (in terms of both the percentage of single-beat segmentations and the magnitude of the annotation errors) compared with models which assume the observations within each state are i.i.d.

6. In autoregressive modeling, the signal sample at time $t$ is considered to be a linear combination of a number of previous signal samples plus an additive noise term. Specifically, an AR model of order $m$ is given by $x_t = \sum_{i=1}^{m} c_i x_{t-i} + \epsilon_t$, where $c_i$ are the AR model coefficients and $\epsilon_t$ can be viewed as a random residual noise term at each time step.
Table 11.1 Five-Fold Cross-Validation Results for HMMs Trained on the Raw ECG Signal Data from Leads II and V2

| Hidden Markov Model Specification | Lead II | Mean Absolute Errors (ms) | | | |
|-----------------------------------|--------|---------------------------|---|---|---|---|
|                                   | % of Single-Beat Segmentations | P\text{on} | Q | J | T\text{off} | |
| Standard HMM                      | 5.7%   | 175.3 | 108.0 | 99.0 | 243.7 | |
| Gaussian observation model        |        |        |        |        |        | |
| No normalization                  |        |        |        |        |        | |
| Standard HMM                      | 69.8%  | 485.0 | 35.8  | 73.8 | 338.4 | |
| Gaussian observation model        |        |        |        |        |        | |
| Range normalization               |        |        |        |        |        | |
| Standard HMM                      | 57.5%  | 272.9 | 48.7  | 75.6 | 326.1 | |
| GMM observation model             |        |        |        |        |        | |
| Range normalization               |        |        |        |        |        | |
| Standard HMM                      | 71.7%  | 49.2  | 10.3  | 12.5 | 52.8  | |
| AR observation model              |        |        |        |        |        | |
| Range normalization               |        |        |        |        |        | |

<table>
<thead>
<tr>
<th>Lead V2</th>
<th>% of Single-Beat Segmentations</th>
<th>Mean Absolute Errors (ms)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard HMM</td>
<td>33.6%</td>
<td>211.5</td>
<td>14.5</td>
<td>20.7</td>
<td>31.5</td>
<td></td>
</tr>
<tr>
<td>Gaussian observation model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No normalization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>77.9%</td>
<td>293.1</td>
<td>49.2</td>
<td>50.7</td>
<td>278.5</td>
<td></td>
</tr>
<tr>
<td>Gaussian observation model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range normalization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>57.4%</td>
<td>255.2</td>
<td>49.9</td>
<td>65.0</td>
<td>249.5</td>
<td></td>
</tr>
<tr>
<td>GMM observation model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range normalization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>87.7%</td>
<td>43.4</td>
<td>5.4</td>
<td>7.6</td>
<td>32.4</td>
<td></td>
</tr>
<tr>
<td>AR observation model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range normalization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Despite the advantages offered by AR observation models, the mean annotation errors for the associated HMMs are still considerably larger than the inter-analyst variability present in the data set annotations. In particular, the T wave offset annotation errors for leads II and V2 are 52.8 ms and 32.4 ms, respectively. This “level of accuracy” is not sufficient to enable the trained model to be used as an effective means for automated ECG interval analysis in practice.

The fundamental problem with developing HMMs based on the raw ECG signal data is that the state observation models must be flexible enough to capture the statistical characteristics governing the overall shape of each of the ECG waveform features. Although AR observation models provide a first step in this direction, these models are not ideally suited to representing the waveform features of the ECG. In particular, it is unlikely that a single AR model can successfully represent the statistical dependencies across whole waveform features for a range of ECGs.
Thus, it may be advantageous to utilize multiple AR models (each with a separate model order) to represent the different regions of each ECG waveform feature.

An alternative approach to overcoming the i.i.d. assumption within each HMM state is to encode information from “neighboring” signal samples into the representation of the signal itself. More precisely, each individual signal sample is transformed to a vector of transform coefficients which captures (approximately) the shape of the signal within a given region of the sample itself. This new representation can then be used as the basis for training a hidden Markov model, using any of the standard observation models previously described. We now consider the utility of this approach for automated ECG interval analysis.

11.8 Wavelet Encoding of the ECG

11.8.1 Wavelet Transforms

Wavelets are a class of functions that possess compact support and form a basis for all finite energy signals. They are able to capture the nonstationary spectral characteristics of a signal by decomposing it over a set of atoms which are localized in both time and frequency. These atoms are generated by scaling and translating a single mother wavelet.

The most popular wavelet transform algorithm is the discrete wavelet transform (DWT), which uses the set of dyadic scales (i.e., those based on powers of two) and translates of the mother wavelet to form an orthonormal basis for signal analysis. The DWT is therefore most suited to applications such as data compression where a compact description of a signal is required. An alternative transform is derived by allowing the translation parameter to vary continuously, whilst restricting the scale parameter to a dyadic scale (thus, the set of time-frequency atoms now forms a frame). This leads to the undecimated wavelet transform (UWT),7 which for a signal $s \in L^2(\mathbb{R})$, is given by

$$w_\nu(\tau) = \frac{1}{\sqrt{\nu}} \int_{-\infty}^{+\infty} s(t) \psi^* \left( \frac{t-\tau}{\nu} \right) dt \quad \nu = 2^k, k \in \mathbb{Z}, \tau \in \mathbb{R} \quad (11.22)$$

where $w_\nu(\tau)$ are the UWT coefficients at scale $\nu$ and shift $\tau$, and $\psi^*$ is the complex conjugate of the mother wavelet.

In practice the UWT for a signal of length $N$ can be computed in $O$ using an efficient filter bank structure [27]. Figure 11.7 shows a schematic illustration of the UWT filter bank algorithm, where $h$ and $g$ represent the lowpass and highpass “conjugate mirror filters” for each level of the UWT decomposition.

The UWT is particularly well suited to ECG interval analysis as it provides a time-frequency description of the ECG signal on a sample-by-sample basis. In addition, the UWT coefficients are translation-invariant (unlike the DWT coefficients), which is important for pattern recognition applications.

7. The undecimated wavelet transform is also known as the stationary wavelet transform and the translation-invariant wavelet transform.
11.8.2 HMMs with Wavelet-Encoded ECG

In our experiments we found that the Coiflet wavelet with two vanishing moments resulted in the best overall segmentation performance. Figure 11.8 shows the squared magnitude responses for the lowpass, bandpass, and highpass filters associated with this wavelet (which is commonly known as the coifl wavelet).

In order to use the UWT for ECG encoding, the UWT wavelet coefficients from levels 1 to 7 were used to form a seven-dimensional encoding for each ECG signal. Table 11.2 shows the five-fold cross-validation results for HMMs trained on ECG waveforms from leads II and V2 which had been encoded in this manner (using range normalization prior to the encoding).

The results presented in Table 11.2 clearly demonstrate the considerable performance improvement of HMMs trained with the UWT encoding (albeit at the expense of a relatively low percentage of single-beat segmentations), compared with similar models trained using the raw ECG time series. In particular, the $Q$ and $T_{off}$ single-beat segmentation errors of 5.5 ms and 12.4 ms for lead II, and 3.3 ms and 9.5 ms for lead V2, are significantly better than the corresponding errors for the HMM with an autoregressive observation model.

Despite the performance improvement gained from the use of wavelet methods with hidden Markov models, the models still suffer from the problem of double-beat segmentations. In the following section we consider a modification to the HMM architecture in order to overcome this problem. In particular, we make use of the knowledge that the double-beat segmentations are characterized by the model inferring a number of states with a duration that is much shorter than the minimum state duration observed with real ECG signals. This observation leads on to the subject of duration constraints for hidden Markov models.

11.9 Duration Modeling for Robust Segmentations

A significant limitation of the standard HMM is the manner in which it models state durations. For a given state $i$ with self-transition coefficient $a_{ii}$, the probability mass
Figure 11.8  Squared magnitude responses of the highpass, bandpass, and lowpass filters associated with the coif1 wavelet (and associated scaling function) over a range of different levels of the undecimated wavelet transform.
Table 11.2 Five-Fold Cross-Validation Results for HMMs Trained on the Wavelet-Encoded ECG Signal Data from Leads II and V2

<table>
<thead>
<tr>
<th>León II</th>
<th>Hidden Markov Model</th>
<th>% of Single-Beat Segments</th>
<th>Mean Absolute Errors (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specification</td>
<td>P on Q J T off</td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>Gaussian observation model</td>
<td>29.2%</td>
<td>26.1 3.7 5.0 26.8</td>
</tr>
<tr>
<td>UWT encoding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>GMM observation model</td>
<td>26.4%</td>
<td>12.9 5.5 9.6 12.4</td>
</tr>
<tr>
<td>UWT encoding</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lead V2</th>
<th>Hidden Markov Model</th>
<th>% of Single-Beat Segments</th>
<th>Mean Absolute Errors (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specification</td>
<td>P on Q J T off</td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>Gaussian observation model</td>
<td>73.0%</td>
<td>20.0 4.1 8.7 15.8</td>
</tr>
<tr>
<td>UWT encoding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard HMM</td>
<td>GMM observation model</td>
<td>59.0%</td>
<td>9.9 3.3 5.9 9.5</td>
</tr>
<tr>
<td>UWT encoding</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The encodings are derived from the seven-dimensional coifl wavelet coefficients resulting from a level 7 UWT decomposition of each ECG signal. In each case range normalization was used prior to the encoding.

The function for the state duration \( d \) is a geometric distribution, given by

\[
p_i(d) = (a_{ii})^{d-1}(1 - a_{ii})
\]

For the waveform features of the ECG signal, this geometric distribution is inappropriate. In particular, the distribution naturally favors state sequences of a very short duration. Conversely, real-world ECG waveform features do not occur for arbitrarily short durations, and there is typically a minimum duration for each of the ECG features. In practice this “mismatch” between the statistical properties of the model and those of the ECG results in unreliable “double-beat” segmentations, as discussed previously in Section 11.7.3.

Unfortunately, double-beat segmentations can significantly impact upon the reliability of the automated QT interval measurements produced by the model. Thus, in order to make use of the model for automated QT interval analysis, the robustness of the segmentation process must be improved. This can be achieved by incorporating duration constraints into the HMM architecture. Each duration constraint takes the form of a number specifying the minimum duration for a particular state in the model. For example, the duration constraint for the T wave state is simply the minimum possible duration (in samples) for a T wave. Such values can be estimated in practice by examining the durations of the waveform features for a large number of annotated ECG waveforms.

Once the duration constraints have been chosen, they are incorporated into the model in the following manner: For each state \( k \) with a minimum duration of \( d_{\text{min}}(k) \), we augment the model with \( d_{\text{min}}(k) - 1 \) additional states directly preceding...
11.9 Duration Modeling for Robust Segmentations

The original state \( k \). Each additional state has a self-transition probability of zero, and a probability of one of transitioning to the state to its right. Thus, taken together, these states form a simple left-right Markov chain, where each state in the chain is only occupied for at most one time sample (during any run through the chain).

The most important feature of this chain is that the parameters of the observation density for each state are identical to the corresponding parameters of the original state \( k \) (this is known as “tying”). Thus the observations associated with the \( d_{\min} \) states identified with a particular waveform feature are governed by a single set of parameters (which is shared by all \( d_{\min} \) states). The overall procedure for incorporating duration constraints into the HMM architecture is illustrated graphically in Figure 11.9.

Table 11.3 shows the five-fold cross-validation results for a hidden Markov model with built-in duration constraints. For each fold of the cross-validation procedure, the minimum state duration \( d_{\min}(k) \) was calculated as 80% of the minimum duration present in the annotated training data for each particular state. The set of duration constraints were then incorporated into the HMM architecture and the resulting model was trained in a supervised fashion.
The results demonstrate that the duration constrained HMM eliminates the problem of double-beat segmentations. In addition, the annotation errors for leads II are of a comparable standard to the best results presented for the single-beat segmentations only in the previous section.

11.10 Conclusions

In this chapter we have focused on the two core issues in utilizing a probabilistic modeling approach for the task of automated ECG interval analysis: the choice of representation for the ECG signal and the choice of model for the segmentation. We have demonstrated that wavelet methods, and in particular the undecimated wavelet transform, can be used to generate an encoding of the ECG which is tuned to the unique spectral characteristics of the ECG waveform features. With this representation the performance of the models on new unseen ECG waveforms is significantly better than similar models trained on the raw time series data. We have also shown that the robustness of the segmentation process can be improved through the use of state duration constraints with hidden Markov models. With these models the robustness of the resulting segmentations is considerably improved.

A key advantage of probabilistic modeling over traditional techniques for ECG segmentation is the ability of the model to generate a statistical confidence measure in its analysis of a given ECG waveform. As discussed previously in Section 11.3, current automated ECG interval analysis systems are unable to differentiate between normal ECG waveforms (for which the automated annotations are generally reliable) and abnormal or unusual ECG waveforms (for which the automated annotations are frequently unreliable). By utilizing a confidence-based approach to automated ECG interval analysis, however, we can automatically highlight those waveforms which are least suitable to analysis by machine (and thus most in need of analysis by a human expert). This strategy therefore provides an effective way to combine the twin advantages of manual and automated ECG interval analysis [3].

References