Topics on Statistical Theory and Applications

by

Heng Lian

B.S., Mathematics, University of Science and Technology of China, 2000
Sc.M., Computer Science, Brown university, 2005
M.A., Economics, Brown University, 2005

Thesis

Submitted in partial fulfillment of the requirements for
the Degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

Ph.D. Advisor: Charles E. Lawrence and Donald E. McClure

PROVIDENCE, RHODE ISLAND

April 2007

The Vita of Heng Lian

Heng Lian was born in the city of Fuzhou, Fujian province on Nov 14, 1977. He started his undergraduate study at the University of Science and Technology of China in Hefei, Anhui in 1996 and graduated in 2000 with B.S. in Mathematics and Computer Science. After a short break, he came to the Unites States and attended graduate school at Brown University in the Division of Applied Mathematics. He studied under the supervision of Professor Donald McClure and Professor Charles Lawrence. While at Brown, he received a Sc.M in Computer Science and a M.A in Economics in 2005. This thesis was defended on April 5th, 2007.

Abstract of "Topics on Statistical Theory and Applications," by Heng Lian, Ph.D., Brown University, April 2007

This thesis includes four chapters, each dealing with a different topic of statistics.

In Chapter 1, after introducing the biological problem that serves as the motivation for our work, we present the detail of our model for sequence segmentation. As an extension of the well-known Hidden Markov Model (HMM), our model is a hierarchical Bayesian model which uses a continuous mixture of Gaussian distributions to model the variations within each state. Detailed and efficient algorithm and estimation procedure based on dynamic programming will be presented after the model is set up. We estimate the hyperparameters using the empirical Bayesian approach aided by stochastic Expectation Maximization(EM) algorithm. The number of states can be estimated by computing the data likelihood conditioned on the estimated hyperparameters. We applied our model to the biological problem of delineating regions of human genome with different levels of sensitivity. After that, we will present some variants and extensions of our hierarchical model. We first give a simpler model that is easier to estimate. Then we present a fully Bayesian approach for model selection, where a procedure similar to sequential importance sampling (SIS) is proposed for the updating of the posterior for the hyperparameters, which is more efficient than brute-force importance sampling. The possibility of extending the model to take into account Markov dependence is discussed.

In Chapter 2, we study the image superresolution problem using a variational approach. In our new approach, we use a formulation inspired by works in dimension reduction in the machine learning community. We approximate each pixel value by a linear combination of its neighbors, based on the provided low-resolution image, and extrapolate the pixel values in the high-resolution image. This new algorithm is demonstrated to better preserve the structure of the image by preventing oversmoothing as usually happens in other approaches.

In Chapter 3, we make some contributions to the theory of consistency and rates of convergence of the nonparametric Bayesian models. In the first part of this chapter, we study the consistency of posterior distribution in estimating step functions. This requires a slight extension of the existing theory to the case with observations that are non i.i.d. In the second part of the chapter, we give a simple proof of rates of convergence of posterior distribution under model misspecification, based on a key identity that has previously been exploited only in the well-specified case.

In Chapter 4, we extend the theory of reproducing kernel Hilbert spaces in order to deal with functional data models with functional response. This theory provides a new framework for functional regression that has clearly better performance in dealing with nonlinear models. We also show that our new estimator performs better than other simplistic ones such as the kernel regression estimate.

Contents

1 Hierarchical Change Point Model for Tiled-array Sequence Segmentation **1**

1.1 Introduction to the Segmentation Problem 2

1.2 Basic Hierarchical Bayesian Model 10

1.2.1 A hierarchical model 10

1.2.2 Forward recursion and backward sampling 16

1.2.3 Empirical Bayes 24

1.2.4 Stochastic Expectation-Maximization 26

1.2.5 Why I do not see stochastic EM converge? 29

1.2.6 Yet another stochastic EM 33

1.2.7 Missing observations 36

1.2.8 Numerics 37

1.2.9 Model selection 39

1.2.10 Connections with GHMM 41

1.2.11 Analysis of computational complexity 43

1.2.12 Summary of the basic model 45

1.3 Application to Biological Data 46

1.3.1 Biological background 47

1.3.2 Results 49

1.4 Variants and Extensions of the Basic Model 57

1.4.1 A simpler model . . . 58
1.4.2 Fully Bayesian model selection . . . 60
1.4.3 Calculating marginal probability from posterior samples . . . 67
1.4.4 Markov dependence . . . 68
1.4.5 Multiple arrays . . . 71

2 Variational Image Superresolution 74
2.1 Introduction . . . 75
2.2 local structure estimation . . . 77
2.2.1 Connections to LLE . . . 77
2.2.2 Comparison with Digital TV Filter . . . 78
2.3 super-resolution . . . 79
2.4 experimental result . . . 79
2.5 Conclusion . . . 80

3 Nonparametric Bayesian Asymptotics 84
3.1 Consistency of Bayesian Approximation with Step Functions . . . 87
3.1.1 Introduction . . . 87
3.1.2 Main Results . . . 88
3.1.3 Misspecification of the Prior . . . 93
3.1.4 Changing Priors . . . 95
3.1.5 Conclusions . . . 97
3.2 Rates of Convergence of Posterior under Misspecification . . . 97
3.2.1 Introduction . . . 97
3.2.2 Preliminaries . . . 98
3.2.3 Rates of Convergence . . . 100
3.2.4 Discussion . . . 105

4 Functional Data Analysis with Functional Repreducing Kernel Hilbert Spaces 106

4.1 Introduction 107
4.2 Functional reproducing kernel Hilbert spaces 110
4.3 Models for functional data 112
4.4 Examples 116
4.4.1 Simulation study 116
4.4.2 Application to the weather data 119
4.5 Conclusions 119

5 Conclusions **124**

List of Tables

1.1 Some hyperparameters estimated for a 4-state model, including hyper-mean, transition probability, and the segment length parameters. For the transition probability, the number 1e-6 appears multiple times because this is the minimum transition probability allowed in our implementation. 52

1.2 Some descriptive statistics. 52

4.1 Results for the simulation study 117

List of Figures

1.1 Illustration of a segmentation for a piece of the data, with $k = 5$. Note that δ_0 and δ_1 are both in the closed state, but there is a change point at c_1. Confidence bands are put in the graph to show different means and standard errors for each segment. 13

1.2 A data sequence (dotted) with the posterior mean of the hidden parameter μ (solid) computed from 500 samples 24

1.3 Log likelihood computed with training data excluded, conditioned on the estimated hyperparameters. 50

1.4 AIC (solid) and BIC (dotted) both suggest a 4-state model. 51

1.5 Prediction of our model shown in the UCSC genome browser. The top panel is the raw data, the middle is the centroid, the bottom is the uncertainty. 53

1.6 QQ-plots for the residual vs. Normal of our hierarchical model for (a) data in nonsensitive region (b) data in intermediate sensitive region (c) data in minor HS (d) data in major HS. 54

1.7 QQ-plots for the residual vs. Normal of the standard 4-state HMM for (a) data in nonsensitive region (b) data in intermediate sensitive region (c) data in minor HS (d) data in major HS. 55

1.8 Top: sequence data. Bottom: probability of change. 72

1.9 Times series simulated from model $AR(1)$ with standard Gaussian noise. Top: $\beta = 0.9$. Bottom: $\beta = -0.9$. 73

2.1 The linear coefficients computed in our method and digital TV filter is quite different for a simple image 81
2.2 Comparison of interpolation results. Upper Left: original high-resolution image. Upper Right: bicubic interpolation, PSNR=26.9. Lower Left: our method, PSNR=27.6. Lower Right: digital TV filter, PSNR=26.9. 82
2.3 Comparison of results on fingerprint image. Upper Left: original high-resolution image. Upper Right: bicubic interpolation, PSNR=16.3. Lower Left: our method, PSNR=16.8. Lower Right: digital TV filter, PSNR=16.2. 83

4.1 Comparison of the GCV with the error computed from validation data. The solid curves are the GCV estimates, the dashed curves are the error computed from validation set. The curves are shifted and normalized to show the shape of the curves rather than its absolute magnitudes. 118
4.2 Daily weather data for 35 Canadian stations, the curves plotted here result from using smoothing splines to fit the raw data. 120
4.3 Raw data (points) and predictions (solid) of log precipitation for four weather stations. 121

Chapter 1

Hierarchical Change Point Model for Tiled-array Sequence Segmentation

1.1 Introduction to the Segmentation Problem

Biological advancement in recent years has generated a huge amount of data that poses new challenges for both theoreticians and practitioners in various fields including mathematics, computer science, and engineering. Most traditional methods in statistics and related fields are not able to efficiently deal with this kind of data, either because of the long computation time involved for large data set or, more seriously, because the model do not fit the real-world biological data well.

Biological sequence analysis accounts for a major part of efforts put into computational biology in the past century. Numerous works have been done trying to find biologically meaningful and important signals hidden in a sea of data that is composed of a few discrete symbols. The problems studied include sequence comparison[41], unusual sequence detection[48], motif discovery[83], just to name a few. The merits of these analyses are multifold. These sequences can be coding segments, analysis of which will lead to new discovery of their physical properties which have important implications for gene function. Comparison of coding sequences for unknown genes can give researchers an educated guess of its functionality by relating it to some other known genes. There is also increasing interest in looking at non-coding regions, which usually contain regulatory elements that control the expression level of some nearby genes. Analysis of the repetitive patterns in these sequences will often reveal functional elements in these regions.

Almost all models that have been proposed for sequence analysis require some distributional assumptions for the background in order for the segments of interest to be discriminable. The power of detection algorithms depends critically on the strength of the signal. The stronger the signal, the easier it is to distinguish it from the background. One omniscient assumption underlying most of these models is that the background consists of homogeneous regions, with each letter independent identically distributed from the background model. These assumptions make theoretical analysis and inference tractable. A more general assumption that is sometimes used is

that of Markov dependence, which is more realistic for many biological data. Higher order Markov dependency has been utilized by some which also demonstrates that it leads to a better biological model. Validity of these models are critical in the computations of various statistics, such as the p-value. Unfortunately, all these models, however complicated they are, have one drawback in common. The parameters for the background model are assumed to be fixed for the whole sequence under study. This assumption is reasonable with relatively short sequences because the underlying distribution generally change slowly. As longer sequences are becoming available for study nowadays, especially after the completion of the human genome project (HGP) in 2003, the importance of dealing with inhomogeneity of sequence is being realized by most researchers in this field. The traditional assumptions are insufficient to deal with such long sequences. Traditional assumptions lead to models that is not valid in the present situation, which in turn produces analysis results that are at least inaccurate and can possibly lead to totally incorrect conclusions. Development of new models and inference procedures is an urgent task. We hope this thesis will be one step in this direction.

Although the importance of incorporating inhomogeneity into consideration when building a mathematical model has been widely recognized, to this day, this problem is still not well studied, let alone solved. Nevertheless, some approaches have been proposed to address this problem. For example, in [45], the authors take into account the local variations in the composition of DNA sequences by estimating background model parameters using only a part of data in a window around regions of interest. Using locally adaptive estimate in place of global computation is an important step forward. A major problem of this and other related approaches is the more or less arbitrary decision made in the choice of the window size. A window with a size that is too small will make inferences unduly influenced by data that should not be included in the background model and thus give inaccurate estimates. A window size that is too big will potentially include several regions with different background characteris-

tics that should be considered separately. Although statistical and signal processing literature has provided us with some theoretical analysis of adaptive estimation[61], these procedures either do not provide a computational method or the computational procedure is too complicated and does not perform well in practice.

One promising approach was laid out in [54]. Using Bayesian approach, they proposed a segmentation model that can automatically learn the number of regions that have different underlying characteristics. Aided by an efficient dynamic programming algorithm, they are able to obtain the exact posterior distribution over the number of segments each with its own underlying parameters. Various statistical inference procedures can be carried out with little effort. Our model is basically an extension of this line of thinking.

In the above we only considered sequence data that consist of a finite number of letters. For example, a DNA sequence is composed of four types of nucleotides, A, T, C, and G, a protein sequence consists of about 20 amino acids. For proteins, we can also consider the sequences as consisting of fewer letters by grouping amino acids according to their different characteristics, like hydrophobicity, charge, weight, etc. Study of sequences of this discrete nature played a critical role in 20th century computational biology.

A major technological breakthrough in system biology is the use of microarray for studying gene mechanisms. A microarray is a glass slide which contains thousands of spots imprinted by an accurate machine, which can grab DNA pieces from a sample prepared and print them onto the array. A microarray can contain sample DNA from genes of interest, with each probe containing thousands of cDNA fragments from a single gene, or it can be a tiled array that is imprinted with probes that cover long regions of the genome or even the whole genome. The DNA sample, which has been dyed with some color, is hybridized to the array. In experimental design, more often than not, another sample of DNA is prepared and hybridized to a separate array which acts as a controlled study for comparison. The array is then scanned by a laser

scanner which produces an image that shows the intensity of each probe. If the sample contains a high concentration of DNA fragments that is complimentary to a probe on the array, the image will show a spot with high intensity in the corresponding position. A tiled microarray produces new type of data which we will call generalized sequence data. Instead of a finite number of possible letters, the tiled array produces sequences consisting of intensities or transformed intensities after preprocessing, which should be modeled as a continuous variable. A considerable amount of noise can be introduced into the data due to point tip variation, dye variation, etc. So the data need to be normalized both within a single array and cross different arrays before further analysis. In the controlled experiment, which should ideally consist of data that is presumably homogeneous across the total covered regions, the inhomogeneity of intensities is still present due to underlying biological and experimental variations.

Segmentation of biological sequences is a well-studied problem. An excellent review on this topic can be found in [10]. Here we will briefly discuss Hidden Markov Model (HMM) and change point model since our proposed new approach borrows strength from both of these two models. HMM is arguably the most frequently used model for data that has multiple distinct underlying states. The HMM model assumes that there is an underlying Markov chain for the observed data sequence. All the information contained in the observations is conveyed by this underlying process. HMM is essentially a mixture model where each state represents one component. For an HMM, each observation sequence $\{Y_t\}$ is modeled by the statistical relationship with its underlying state sequences $\{X_t\}$. HMM is an example of the more general Graphical Models. There are two kinds of constraints imposed on those sequences. The first constraint stipulates that $\{X_t\}$ is a Markov chain, i.e. X_t is independent of $X_1, \ldots X_{t-2}$ if X_{t-1} is given. The second constraint requires that given X_t, Y_t is independent of all other variables. Formally, we have the following generative model

by the above constraints:

$$P(\{X_t, Y_t\}_{t=1}^T) = P(X_1) \prod_{t=2}^{T} P(X_t|X_{t-1}) \prod_{t=1}^{T} P(Y_t|X_t)$$

Although the underlying process is constrained to be a Markov chain, the marginal distribution of the observations can have long range dependency. In fact, it has been shown that the HMM is universal in the sense that it is dense in the space of essentially all finite-state discrete-time stationary processes[36]. This result probably explained the many successful applications in various areas.

The type of HMM that concerns us is HMM with continuous observations, where the likelihood model $P(Y|X)$ is a continuous distribution. There are two sets of parameters for an HMM. One for the discrete time, finite state Markov chain, with $D \times D$ transition matrix a_{ij} and usually also prior distribution π_i, where D is the number of states for the Markov chain. The other set of parameters is for the likelihood model, implicitly contained in $P(Y|X)$(emission probability). Here we only consider a normal likelihood model, so the parameters are the mean and variance for each state (μ, σ^2). HMM has been successfully applied in many fields such as speech recognition[84], natural language processing[16], protein structure prediction[17], and numerous new applications are being explored as we speak. There also exist many extensions and variants of HMM, including so-called profile HMM, pair HMM, factorial HMM, population HMM. Extension of HMM to high dimensions is call Hidden Markov Random Field (HMRF). There also exists continuous state HMM generally called state-space model, and even continuous-time HMM used in engineering applications. A particular extension of HMM that will be of interest to us later is the Generalized Hidden Markov Model(GHMM), also called Hidden semi-Markov Model(HSMM). Instead of emitting one observation at each time step, multiple observations are made per state. The number of observations per state is random and can be modeled by a discrete distribution such as a Geometric distribution. The limitation of the Geometric dis-

tribution has led to the development of other duration models, using continuous distribution such as normal or Gamma, which produces more flexible models. Our model has some similarity with the GHMM, and this connection will be spelled out in detail later.

Many algorithms for learning and inference under an HMM are well-known. Maximum likelihood estimation can be applied if training data are available. Traditionally, Baum-Welch algorithm (a particular instantiation of the celebrated EM algorithm) was used for unsupervised model learning. The inference for the hidden states can be achieved with forward-backward algorithm. Finally, a single most probable state sequence can be found with the Viterbi algorithm, a form of dynamic programming not unlike one used in the forward-backward algorithm. These exact inference procedures can be extended to almost all its variants extensions. For more complicated models, very large datasets, or more complex state space, these algorithms are sometimes too slow. In those cases, people have come up with different approximate algorithms. Gibbs sampling[35] and, more generally, the Metropolis-Hasting algorithm are valuable tools for this purpose. Gibbs sampling works by repeatedly picking a site along the sequence (either deterministically scanning through all sites, or picking randomly), and sample the state at this site conditioned on the states for other neighboring sites that was sampled in previous iterations. More recently, variational inference [47] and variational Bayesian inference [3] provide an interesting alternative. These approaches usually proceed by artificially adding independence assumptions to the variables, and the inference is done under this extra assumption which make the problem much more tractable. Although there exist some theoretically proven bounds for this kind of approximation, it is unclear how accurate it is. There is also a trade-off between using simpler models with exact inferences and using complicated models with approximate inferences, this is investigated to some extent in [12].

As we discussed above, sequence data usually exhibit localized characteristics. The implication of this for the mathematical model that we want to build is that

we need to delineate sequence into distinct segments which have different underlying distributions. Also, information that is common to the whole sequence should be shared for efficient inference and estimation. Change point model is usually used to fulfill this mission. Although it is called a change point MODEL by many researchers, it is really a concept that group together many mathematical models that all try to solve a class of similar problems. The underlying assumption of a change point model is typically that there are certain points along the sequence where there is a sudden change of distributions. Specifically, for a sequence $\{X_t\}$, suppose the distribution of X_t changes to a different distribution after c_1 observations, and then changes to yet another distribution after $c_2 - c_1$ more observations, then c_1 and c_2 will both be change points, and observations within one fixed segment $X_{c_1+1} \dots X_{c_2}$ will be assumed to come from the same distribution. Perhaps the simplest case which is also most useful in practice is changes of the distribution mean. It is not hard to imagine more complicated type of changes, like changes in higher order moments, changes in dependence structure, etc. We can even assume nonparametric changes in which we only know that some aspect of the distribution is changed, but no other more concrete information is available.

Many different approaches to the change point problem have been proposed. Many split-and-merge algorithms can be utilized as a global approach. Locally weighted regression [27] developed in the time series literature can be also used with minor changes. If the observations are i.i.d. from the underlying distribution, an HMM is a natural modeling tool. In an HMM, different states can be connected with different emission distributions. If the underlying state changes at a point, the following observations will rise from a different distribution. A contiguous stretch of subsequence with the same underlying state will be considered as a homogeneous segment.

One implication of the Markov assumptions of the HMM is that the length of a contiguous segment within the same state is geometrically distributed, since at each time step, the chain has a fix probability of staying at the same state as in the

previous time step, a short-memory property characteristic of the Geometric distribution. Using geometrically distributed random variables for modeling durations in this kind of situations is largely perceived as a convenience for inference rather than a reality-based modeling choice. Real world sequence data with known segmentations rarely exhibit the geometric length distribution. GHMM can solve this problem by explicitly modeling the length with certain more flexible distributions such as normal or Gamma discussed above, with some increase in computation time. Dynamic programming-like recursion for GHMM is typically an $O(n^2)$ algorithm, quadratic in the length of the sequence, whereas the computational complexity of HMM increases only linearly with sequence length. Note that the continuous counterpart of the geometric distribution, the exponential distribution, is a special case of Gamma, so the Gamma distribution is really a generalization of the Geometric distribution in some sense. GHMM has been widely used in speech recognition and biological sequence analysis, but it has another problem in common with HMM.

Although HMM can be used in an abstract context where the state space is perceived only as a finite set that has no ordering or meaning, in biological problems, the investigator usually has some rough a priori belief about how many states should be reasonably considered in a multi-state HMM, using field-specific prior knowledge. The states will typical have some biological interpretation, rather than just a symbol used to distinguish one from the other. For example, they might prefer to assign one state to the part of the sequence that is of interest, and another state to model the background sequence. More states can be assigned to both the foreground and the background model to distinguish subsequences with a finer scale. HMM is built on the hypothesis that each state has a fixed distribution attached to it. As discussed above, since inhomogeneity is the norm rather than exception in sequence data, a simple distribution like a normal distribution with state-specific mean and variance cannot accommodate such variations within the state. In order to achieve a better fit for the data at hand, many more states are typically required, which not only

increases considerably the computation time needed, but more seriously, the model becomes difficult to interpret. Even if it can fit the data closely, biologists will find such kind of model unappealing. Another worry one may have is that allowing so many states will tend to overfit the data, which will produce a model that has low generalization capacity. A conceptually viable approach is using a more complicated emission distribution other than normal distribution for each state. Complicated distributions lead to complicated estimation problems. When dealing with the situation that the normal model is inadequate, which is a recurring issue in applied statistics, a cure-all solution favored by some is to replace the normal distribution by a finite mixture of normal distributions. Using this in HMM is equivalent to increasing the number of states, which is not a satisfactory solution as discussed above. We propose a Bayesian model that can be regarded as a continuous mixture of normal distributions. By exploring the conjugacy property for standard parametric distributions, inference and estimation problems using our model can be efficiently solved.

We also realize that the characteristics of a sequence usually change gradually, so a model that specifies a specific point of change might not be appropriate depending on the applications. One would prefer to have a model that shows a gradual change in the sequence, instead of abrupt occurrence of events. We will return to this point in Section 1.3, where we argue that this problem is at least alleviated by our inference procedure.

1.2 Basic Hierarchical Bayesian Model

1.2.1 A hierarchical model

We model the data as observations from a multi-state Markov chain. In the usual HMM approach, each observation with a given state is modeled as being drawn from a specific distribution, most frequently a normal distribution. As illustrated in the

next section we find that using this simple approach does not fit our data well. To address this poor fit, we designed a hierarchical model that uses a continuous mixture of normals to model each state, that is, we put a prior on the mean and variance of the observations given the state. Within each state, there can be different means and variances for different parts of the data. It turns out that the hidden parameters consisting of means and variances of our observations can be integrated out analytically, as well as the location of the change points. This observation means our model has roughly the same computational complexity as the GHMM, which is quadratic in the length of the sequence.

Our model is based on the model in [54]. The modifications and extensions we made to that model include the following:

1. We extend the model to the continuous case where observations are modeled as continuous random variables. This change essentially imposes no extra technical difficulty, if conjugate priors are used as in the discrete model, which makes explicit integration possible.

2. We embed the segmentation model in [54] into the traditional HMM model, and build a new hierarchical Bayesian model. So our segmentation model is comprised of a hierarchical structure where neighboring segments are first organized into contiguous pieces with the same state, and observations arise from the concatenation of segments from different states.

3. In [54], the hyperparameters are assumed to be given, which may be unsatisfactory from a statistical point of view. We propose a principled method to estimate these hyperparameters within the empirical Bayes (EB) paradigm combined with a stochastic version of the EM algorithm. We show that the essential steps required by the EM algorithm is made possible by utilizing the forward recursion and backward sampling approach.

4. Data with incomplete observations are frequently encountered in practice. The

hidden variables can also be considered as missing data. In fact, a large amount of work has been done to address the difficulty in inference arising from the existence of hidden variables, which are regarded as missing data, including the EM algorithm which is specifically designed for this purpose. In this work, we use missing data to refer to some regions of the sequence that is purely unobserved, producing gaps in our data. Many well-known more or less complicated algorithms like imputation can be used to solve this problem. We show that within our framework, the inferences can proceed without recourse to the imputation procedure. This is done by using the most common trick of Bayesian inference — integrate as much as you can to get rid of all variables that you consider as a nuisance. In contrast, in almost all previous biological applications, the most frequently adopted strategy is to just ignore the missing observations.

5. Instead of using one model with a prespecified number of states, we use the data itself to help us select the appropriate number of states. For the problem that will be described in the next section, this is important since the biologists do not agree on how many biologically meaningful states there should be in the data.

Specifically, the model can be described as follows. Let us assume for now that we know the distribution on the length of each segment, for each of the states, and we know the transition probabilities between the states. Suppose the maximum number of change points is k_{max}. A particular segmentation can be denoted by a vector $A = \{\delta_0, c_1, \delta_1, c_2, \ldots, \delta_{k-1}, c_k, \delta_k\}$, $1 \leq c_1 < c_2 < \ldots < c_k < n$, where n is the length of the sequence, c_i is the ith change point, and δ_i is the state between its neighboring change points taking value in the set of possible states $\{1, \ldots, D\}$. For our application in the next section, the states are actually ordered with 1 representing the state with the smallest mean and D representing states with the largest mean. For our discussion now, these states have no concrete meanings, and may well be described with other symbols instead of integers. Note that by convention, for two

neighboring segments, the last observation of the first segment will be considered as the change point between the two, so the state at the change point c_i is δ_{i-1}. Figure

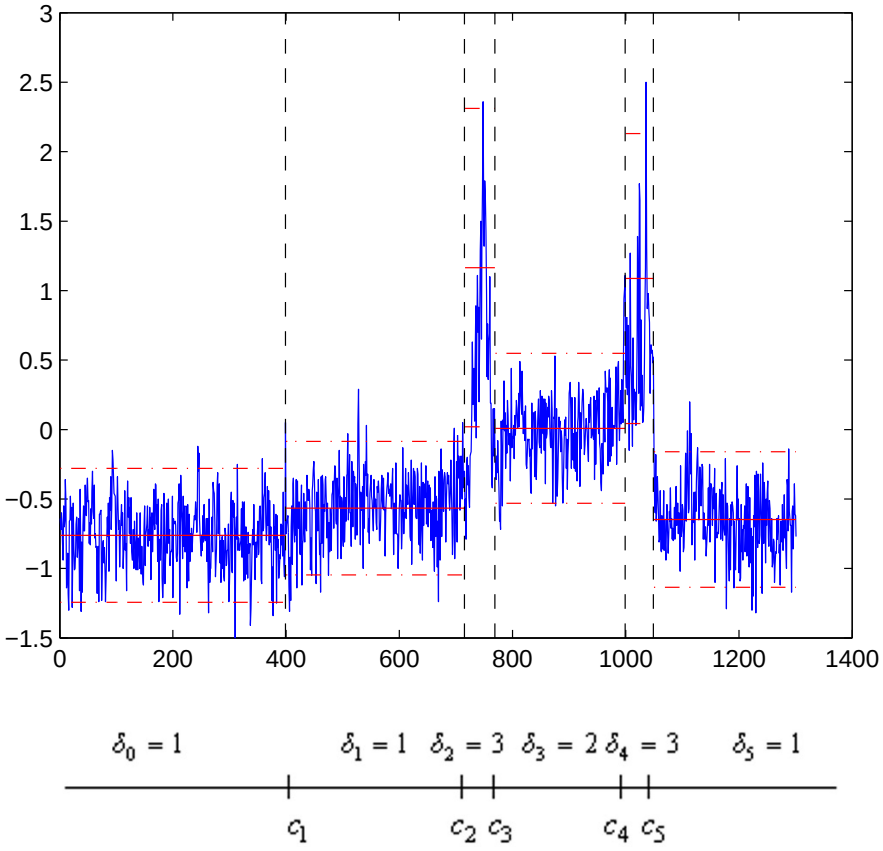


Figure 1.1: Illustration of a segmentation for a piece of the data, with $k = 5$. Note that δ_0 and δ_1 are both in the closed state, but there is a change point at c_1. Confidence bands are put in the graph to show different means and standard errors for each segment.

1.1 shows an example with five change points $\{c_i\}_{i=1}^5$ in which the first two substrings are found to be in state 1, but these two adjacent substrings each has its own mean (solid lines) and variance (dash-dotted lines). As the figure illustrates, the means of the substrings in state 1 are substantially lower than those in state 2, which are in turn much lower than those of state 3. The recursion marginalizing over change points and transitions which we will present in the following is an extension of the recursion used by Liu and Lawrence (1999)[54], but adds state transition terms like

those in HMMs. It differs from an HMM in that it does not permit state transitions at every observation but only at the change points. In turn, the change points can be anywhere and are also inferred from the data. Marginalization over substring means and variances is achieved by integrating over these in the hierarchical models. These marginalization steps greatly reduce dimensionality of the space of unknowns, and thus substantially reduce the potential for over-fitting.

If we assume that the state assignment for all segments is a Markov chain, then the prior probability of A before observing the data is

$$p(A) = \prod_{i=0}^{k} p_{\delta_i}(c_{i+1} - c_i + 1) \cdot \pi(\delta_0) \prod_{i=0}^{k-1} K(\delta_i, \delta_{i+1})$$

where $p_\delta(l)$ is the probability of the length of one segment being l, given the segment state is δ, K is the transition probability between states with initial distribution π. Note that we put transition probabilities between segments, instead of between each data point as in the traditional HMM. From another point of view, this is same as assuming multiple observations per state, as in GHMM. Also, the transitions between the same state is allowed. In a GHMM, it does not make too much sense to allow transitions that stays in the same state, although this point is rarely mentioned, and GHMM usually proceed without this constraint, probably just for convenience. Larger units consisting of one or more adjacent segments with the same state are called "fragments". We set $c_0 = 0$ and let c_{k+1} be the last position in the sequence. From now on, for clarity of exposition, we always consider c_{k+1} to be the last change point of the sequence, so the number of change point is the same as the number of segments, and we will not distinguish between these two.

Given the segmentation A, the mean and variance for each segment is generated

from $normal - inverse - \chi^2$ distribution

$$\mu_i|\sigma_i^2, A \sim N(\mu_{\delta_i}^{(h)}, \frac{\sigma_i^2}{k_{\delta_i}^{(h)}}) \tag{1.1}$$
$$\sigma_i^2|A \sim Inv\chi^2(\nu_{\delta_i}^{(h)}, \sigma_{\delta_i}^{2\,(h)})$$

Those parameters with superscript (h) are hyperparameters that need to be specified or estimated from the data. We will refer to the μ and σ in the above as hidden parameters, and the segmentation A itself as the hidden variable. Each of the segments has a different mean and variance associated with it, and these mean and variances are independently drawn from one of D priors. Which prior to draw from is determined by the state of the segment. The individual mean and variance for each segment is used to model the variations in different segments, even when they come from the same state. The shared information within one state is contained in the priors that are used for the whole sequence.

With segmentation A and μ_i, σ_i given for each segment, the observations are naturally modeled as normal with given mean and variance:

$$y_{c_i+1:c_{i+1}}|\mu_i, \sigma_i^2, A \overset{iid}{\sim} N(\mu_i, \sigma_i^2) \tag{1.2}$$

where we used the notation $y_{i:j} = \{y_k|k = i, i+1, \ldots, j\}$. Note (1.1) and (1.2) are exactly the conjugate pairs that is used in [34] which make analytic integration possible.

In summary, our Bayesian model can be described as follows. First, we have one prior for each state on the hidden parameters — the mean and variance for each segment. We also have prior on the possible segmentations determined by the length distribution and state transition probabilities between segments. Hidden parameters for each segment are drawn independently from the prior conditioned on the segment state. Finally, the observations are generated independently from the normal

distribution for each segment.

We model the segment length using the $Gamma(\alpha, \beta)$ distribution, where α is the shape parameter, and β is the scale parameter. Since different parameterizations are used by different researchers, to clarify the notation, in our parameterization the mean of the $Gamma(\alpha, \beta)$ distribution is α/β. Although this distribution is a little bit more complicated to handle than the normal model, it is usually believed to be a more appropriate model for length distribution, and there is no extra problem caused by the negative value that can be taken by a normal distribution which may need to be truncated. The hyperparameters are $\Theta = \{\mu_\delta^{(h)}, k_\delta^{(h)}, \nu_\delta^{(h)}, \sigma_\delta^{2(h)}, K(\delta, \delta'), \pi(\delta), \alpha_\delta, \beta_\delta : \delta, \delta' \in \{1, 2, \ldots, D\}\}$. The hyperparameters have two separate parts, $\Theta_1 = \{\mu_\delta^{(h)}, k_\delta^{(h)}, \nu_\delta^{(h)}, \sigma_\delta^{2(h)}\}$ contains those used in the prior specification of hidden parameters representing mean and variance for each segment, the other part $\Theta_2 = \{K(\delta, \delta'), \pi(\delta), \alpha_\delta, \beta_\delta\}$ is associated with priors on the change point locations and state assignments. For now, we assume these hyperparameters are known, then the main goal of inference is to get at the posterior distribution $P(A|y)$. Here as well as in the following, we usually omit explicitly writing the dependence of the probability distribution on the hyperparameters Θ. We show the difficulty of computing the posterior distribution above in the next subsection and explain a dynamic programming recursion for solving it.

1.2.2 Forward recursion and backward sampling

By the Bayes formula, we have $P(A|y) = p(y|A)p(A)/\sum_{A'} p(y|A)p(A)$, where $p(y|A)$ is the likelihood in which the hidden parameters are integrated out. By the independence of y in different segments conditioned on the hidden parameters, we have

$$p(y|A) = \prod_{i=0}^{|A|-1} p(y_{c_i+1:c_{i+1}}|\delta_i)$$

where $|A|$ is the number of segments in the segmentation A. Each term in the product can be analytically evaluated in thanks to the conjugate prior used (here superscript (h) is omitted in the hyperparameters)

$$\begin{aligned} p(y_{a:b}|\delta) &= \int\int \frac{1}{\sqrt{2\pi\sigma^2}^{b-a+1}} e^{-\frac{\sum_i (y_i-\mu)^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi\frac{\sigma^2}{k_\delta}}} e^{-\frac{k_\delta(\mu-\mu_\delta)^2}{2\sigma^2}} \cdot \\ &\quad \frac{(\frac{\nu_\delta}{2})^{\nu_\delta/2}}{\Gamma(\frac{\nu_\delta}{2})} \sigma_\delta^{\nu_\delta} \sigma^{-2(\frac{\nu_\delta}{2}+1)} e^{-\frac{\nu_\delta\sigma_\delta^2}{2\sigma^2}} d\mu d\sigma^2 \end{aligned}$$

We first compute the integration over μ

$$\begin{aligned} & \int e^{-\frac{\sum_i (y_i-\mu)^2}{2\sigma^2}} e^{-\frac{k_\delta(\mu-\mu_\delta)^2}{2\sigma^2}} d\mu \\ = & \sqrt{2\pi\frac{\sigma^2}{k_\delta+b-a+1}} e^{-\frac{\lambda}{2\sigma^2}} \end{aligned}$$

where

$$\lambda = k_\delta\mu_\delta^2 + (b-a+1)\bar{y}^2 - \frac{(k_\delta\mu_\delta + \sum_i y_i)^2}{k_\delta+b-a+1} + \sum (y_i-\bar{y})^2$$

After some tedious calculation, we get

$$p(y_{a:b}|\delta) = \frac{1}{(2\pi)^{(b-a+1)/2}} \sqrt{\frac{k_\delta}{k_\delta+b-a+1}} \frac{g(\nu_\delta,\sigma_\delta^2)}{g(\nu_\delta+b-a+1, \frac{\nu_\delta\sigma_\delta^2+\lambda}{\nu_\delta+b-a+1})} \tag{1.3}$$

where the function g is defined by

$$g(x,y) = \frac{(\frac{x}{2})^{\frac{x}{2}}}{\Gamma(\frac{x}{2})} y^{x/2}$$

The part that gives us trouble in computing $p(A|y)$ is the summation over all different segmentations. Evaluation of this sum in a brute force way is computationally prohibitive since the number of possible segmentations increases with n like $n^{k_{max}}$. This is computationally intensive with even a sequence of reasonable length n with typical choice of k_{max}. This sum is sometimes called partition function which is

the normalizing constant in general graphical models. Approximate computation of the sum can be achieved by sampling schemes that does not require knowledge of the normalizing constant. Importance sampling (IS) or Markov Chain Monte Carlo (MCMC) can be used, but it is difficult to come up with proposal distributions that is efficient enough for our purpose due to the high dimensionality of A. Fortunately, a dynamic programming recursion similar to the one used in [54] can be used to reduce the complexity to $O(n^2 k_{max})$ as we explain in the following.

Let $p(i, \delta, k)$ be the probability that the first k segments has a total length i and the last segment of the subsequence of length i is in state δ . These probabilities can be computed using recursion. If the kth change point is at position i, the previous change point can be at any position $j < i$, with possible state δ'. So we have the following recursion:

$$\begin{aligned} p(i, \delta, 1) &= \pi_\delta p_\delta(i) \\ p(i, \delta, k) &= \sum_{j<i, \delta'} p(j, \delta', k-1) p_\delta(i-j) K(\delta', \delta) \end{aligned} \tag{1.4}$$

The main recursion can be computed as follows:

$$\begin{aligned} & Pr(y_{1:i} | y_{1:i} \text{ has } k \text{ change points } (k \text{ segments}), \text{ with last state } \delta) \\ = & \sum_{j<i, \delta'} Pr(y_{1:i}, \text{ previous change point is at } j, \text{ the state ending at } j \text{ is } \delta' | \delta, k) \\ = & \sum_{j<i, \delta'} P(j, \delta', k-1 | i, \delta, k) P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \\ = & \sum_{j<i, \delta'} \frac{p(j, \delta', k-1) p_\delta(i-j) K(\delta', \delta)}{p(i, \delta, k)} P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \end{aligned} \tag{1.5}$$

where on the third line above we used the independence of observations for different segments given the segmentation. All of the probabilities above should be understood as being conditioned on the hyperparameters, which we omit for simplicity in notation.

The base case when $k = 1$ for the above recursion can be obtained by integrating

out the hidden variables:

$$P(y_{i:j}|\delta,1) = \int P(y_{i:j}|\mu,\sigma^2)p(\mu,\sigma^2|\mu_\delta^{(h)},k_\delta^{(h)},\nu_\delta^{(h)},\sigma_\delta^{2(h)})\,d\mu d\sigma^2 \tag{1.6}$$

The integration can be done analytically since we used the usual conjugate prior in the model. In fact, we have already computed this in equation (1.3).

A very similar recursion was used in [54]. The main difference is that they did not assume an explicit length distribution, and all segmentations with same number of segments are a priori equally likely. Also, they did not assign explicit states for the segments, so there is no transition between the segments. This can also be regarded as a single-state model. The extra components in our model cause additional complications.

After $p(y|\delta,k)$ is computed by the recursion, we can compute the inverted probability $p(\delta,k|y)$ using Bayes rule:

$$p(k \text{ segments with last state } \delta|y_{1:n}) \propto p(n,\delta,k)p(y_{1:n}|\delta,k)$$

The posterior probability of the number of segments can be obtained from the above by summing over δ

$$p(k|y) = \sum_\delta p(\delta,k|y)$$

Almost all of the desired probabilities of interest can be easily obtained. For example, the marginal likelihood is just

$$p(y_{1:n}) = \sum_{k,\delta} p(n,\delta,k)p(y_{1:n}|\delta,k)$$

We can compute the marginal probability that a change point will occur at j:

$$\begin{aligned} & p(c_k = j \text{ for some } k|y) \\ = \; & \frac{1}{p(y_{1:n})} \sum_{k,\kappa,\delta,\delta'} p(j,k,\delta) p(n-j, \kappa-k, \delta') p(y_{1:j}|\delta, k) p(y_{j+1:n}|\delta', \kappa-k) \end{aligned} \quad (1.7)$$

Using the marginal likelihood, we can also compute the probability of a specific segmentation by $p(A|y) = p(A)p(y|A)/p(y)$. Note here we just computed the probability of a given segmentation, this does not give us a sense of which A has high probability. The problem is same as before — we simply have too many possible segmentations and it is not feasible to compute the probability for all of them.

However, we can obtain exact and independent samples from the posterior distribution on the segmentations given the data. First we draw a value k for the number of segments together with the last state of segmentation δ from the posterior $p(k, \delta|y)$, which we have derived above. We set $c_k = n$, and $\delta_{k-1} = \delta$. Then we can recursively sample backward from the following distribution:

$$\begin{aligned} & P(c_{k-1} = j, \text{ state ending at } j \text{ being } \delta' | y_{1:n}, c_k = i, \text{ state ending at } i \text{ being } \delta) \\ \propto \; & p(y_{1:j}|\delta', k-1) \cdot p(y_{j+1:i}|\delta, 0) p(j, \delta', k-1) p_\delta(i-j) K(\delta', \delta) \end{aligned} \quad (1.8)$$

The above derivation used Bayes rule, that the conditional distribution at the 1st line is proportional to the joint distribution.

This forward recursion backward sampling procedure is reminiscent of the forward-backward algorithm for HMM, in which the forward step computes the probability of past observations summing over all previous states, and the backward step computes the probability of future observations given the current state. This is also similar to the Viterbi algorithm, where the forward step is used to find the optimal value of the objective function, keeping track of the previous optimal state, and then backtracing is used to find the optimal states that lead to the optimal value.

If a single estimator is desired, the most frequently used one is the maximum a posteriori estimate (MAP). Using dynamic programming, MAP estimate can be obtained by recursion. We let $V(i,\delta,k) = \max P(\delta_0, c_1, \delta_1, ..., c_{k-1}|c_k = i, \delta_{k-1} = \delta, y_{1:n})$, the maximum probability that can be obtained for previous segmentations conditioned on the kth change point and the kth state. Given the location j of the $k-1$th change point and the state δ' of the $k-1$th segment, $V(i,\delta,k)$ is independent of the information contained in the segments $1 \ldots k-2$, since the maximum probability configuration up to position j is summarized by $V(j,\delta',k-1)$. So we have the following recursion:

$$\begin{aligned} V(i,\delta,1) &= 1 \\ V(i,\delta,k) &= \max_{j,\delta'} p(c_{k-1} = j, \delta'|c_k = i, \delta, y) \cdot V(j,\delta',k-1),\ k > 1 \\ \phi(i,\delta,k) &= \arg\max_{j,\delta'} p(c_{k-1} = j, \delta'|c_k = i, \delta, y) \cdot V(j,\delta',k-1),\ k > 1 \end{aligned} \tag{1.9}$$

and the optimal segmentation can be found by backtracing. First the optimal number of change points and the last state are determined by

$$(k,\delta) = \arg\max_{k,\delta} V(k,\delta,n) \cdot p(k,\delta|y_{1:n})$$

Setting $c_k = n, \delta_{k-1} = \delta$, we recursively find the previous segment:

$$(c_{k-1}, \delta_{k-2}) = \phi(c_k, \delta_{k-1}, k)$$

Note the probability $p(c_{k-1} = j, \delta'|c_k = i, \delta, y)$ used in the recursion (1.9) is exactly what we used when sampling back.

Maximum a posteriori estimate takes another $O(n^2)$ computation time after the forward recursion. A simpler method to get at the posterior distribution is to compute an estimate from the samples drawn. Drawing one sample only takes linear time as can be seen from the derivation (1.8) above. Here we use "centroid" deduced

from samples as our estimate. Formally, the centroid is the segmentation that is the minimum sum of certain kind of distance from the sampled segmentations [24]. This requires a metric to be defined on the state space, which can be specified somewhat arbitrarily. The samples we obtain usually has an dominating vote for a single state, which should be the state for the centroid whatever reasonable metric we use. In our case, the centroid can be obtained as follows. We first draw n samples (500 samples used in our implementation), and for each i, we estimate of the state at position i, $\hat{\delta}_i$, to be the state that is agreed upon using majority vote from all samples. This estimate actually achieves the minimum of sum of Hamming distance. In [24], the author also found that using centroid instead of MAP estimate dramatically improves the prediction accuracy for their problem. We do not see much difference between these two estimates for our application in the next section. But it is still advisable to compute both estimates for comparison. If they are found to be different, more study is required to look into how this happens. The difference usually results from high dimensionality and high variability of the underlying structure.

The samples can also be used to approximate the marginal probability of change point locations in equation (1.7). We can simply count what fraction of samples has a change point at certain positions. In the exact formula (1.7), $p(y_{j+1:n}|\delta', \kappa - k)$ is not directly available after forward recursion, which only computed $p(y_{1:i}|\delta, k)$ for the subsequence starting from position 1, so another "backward recursion" will be required to obtain this probability. Thus the sampling approach is much easier to implement.

Another advantage of using samples is that we immediately get a sense of the uncertainty by looking at what percentage of all those n samples voted on the state $\hat{\delta}_i$. We must remind the reader that this percentage only reflect the inference uncertainty. Anyone who does not have confidence in the model itself should be cautious about the result even when all of the samples agree on one single state, producing an uncertainty of 0.

The hidden variables, mean and variance for each segment, can also be estimated. Given a sampled segmentation, the hidden parameters can be estimated for each segment independently. Conditioned on the state δ of one segment with length l, the prior on the hidden parameters is the $normal - inverse - \chi^2$ distribution. The posterior distribution of the mean and variance are well known, they are in the same $normal - inverse - \chi^2$ family with updated parameters [34]:

$$\begin{aligned} \mu &= \frac{k_\delta^{(h)}}{k_\delta^{(h)} + l} + \frac{l}{k_\delta^{(h)} + l}\bar{y} \\ k &= k_\delta^{(h)} + l \\ \nu &= \nu_\delta^{(h)} + l \\ \nu\sigma^2 &= \nu_\delta^{(h)}\sigma_\delta^{2(h)} + \sum_i (y_i - \bar{y})^2 + \frac{k_\delta^{(h)} l}{k_\delta^{(h)} + l}(\bar{y} - \mu_\delta^{(h)})^2 \end{aligned} \tag{1.10}$$

Fix a position i in the sequence. After N samples are drawn, the posterior for hidden parameters at each position is a mixture of N $normal-inverse-\chi^2$ distributions, with each component corresponding to one sample:

$$\frac{1}{N}\sum_{n=1}^{N} N - Inv\chi^2(\mu, \sigma^2 | \mu^{(n)}, k^{(n)}, \nu^{(n)}, \sigma^{2(n)})$$

where $\mu^{(n)}, k^{(n)}, \nu^{(n)}, \sigma^{2(n)}$ are the values computed as in (1.10) using the nth sample. We can use the mean of this mixture distribution as a summary statistics computed from multiple samples. Fig 1.2.2 shows the mean of μ_i, which looks like a smoothed version of the original data. One use of these hidden parameters is to check model fit, as we will do in the next section, where the residual plots clearly show that our hierarchical model corrects the poor fit by HMM to some extent.

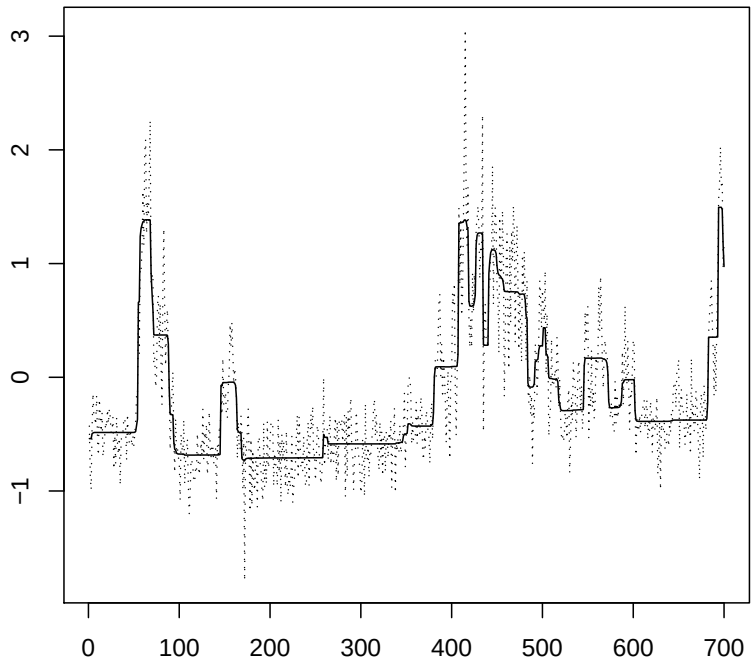


Figure 1.2: A data sequence (dotted) with the posterior mean of the hidden parameter μ (solid) computed from 500 samples

1.2.3 Empirical Bayes

We have up to now assumed that the hyperparameters Θ are fixed in advance. In some simple cases, those hyperparameters can be specified beforehand by an expert who has some idea about the magnitude of these parameters due to previous experience. Although this sounds like a simple task, prior elicitation is the most important and time consuming part of Bayesian analysis. Good priors are usually difficult to obtain, and the poorly chosen prior will usually bias the inference, although this practical issue does not seem to concern theoretical statisticians. In general, empirical Bayes (EB) is a principled method to estimate hyperparameters in a Bayesian model, and we will discuss it in this subsection.

We will give a short review of EB here. Suppose the likelihood model is $p(y|\theta)$ and the prior model is $p(\theta|\lambda)$, where λ is the hyperparameter in the prior. In a fully Bayesian approach, we will use a hyperprior on λ to integrate over this second-level uncertainty. It does not take much thought to realize that this approach will become messy. In a empirical Bayesian approach, the prior is estimated by the data itself. This concept of using data to estimate the prior causes some concern among many Baysians, since prior is used to incorporate informations one has before one actually sees the data, and EB seems to overuse the observed data. We will not go further into this kind of discussion, an excellent exposition on both the philosophical question and theoretical property of EB can be found in [8]. EB works by using the marginal likelihood of y given λ, integrating out θ:

$$p(y|\lambda) = \int p(y|\theta)p(\theta|\lambda)d\theta$$

Then λ can be estimated by maximum likelihood. EB is also called type-II maximum likelihood in the literature. Although the EB framework is simple to describe, there is usually computational difficulty involved in using it in practice.

For our hierarchical Bayesian model, the marginal probability of the sequence is $p(y|\Theta)$, after summing over segmentations and integrating out hidden parameters. This marginal probability is computed by forward recursion in section 1.2.2, which takes $O(n^2)$ time. Although dynamic programming recursion has provided an efficient way to compute this for fixed hyperparameters, this does not solve our problem. If we use some numerical software package to optimize this objective function, a large number of evaluations are probably needed to find the optimum, which is too costly in computation time. Optimization over the high dimensional parameter space (about 40 dimensions for a 4-state model) is generally difficult although not impossible. Also, many efficient numerical optimization algorithms require the gradient as an input, which is also difficult to calculate for our problem. In the following, we will see an approach that decouples the hyperparameters such that the optimization step can be

done either analytically, or numerically in low dimensions.

In a fully Bayesian approach, we would place another prior on those hyperparameters, which is considered to be a more satisfying approach from a Bayesian point of view. But since we have tens of thousands of observations and only dozens of parameters, the maximum likelihood theory suggests that those hyperparameters can be quite accurately estimated by maximizing the likelihood.

1.2.4 Stochastic Expectation-Maximization

As discussed above, we cannot directly perform maximization of the marginal likelihood to get an estimate for the hyperparameters. Directly maximizing the likelihood using some optimization toolbox is impractical for at least two reasons. First, for each fix set of hyperparameters, evaluating the likelihood would require doing the recursion from the start, which is too expensive since the optimization procedure will require many such evaluations. Second, optimization over parameters in high dimensions is difficult especially when the gradient is not easy to compute. A natural approach to avoid this is by using the EM algorithm and regarding the segmentation as the missing variables. The EM algorithm is an efficient method dealing with missing data problems. In the E-step, it computes the sufficient statistics of A in the complete model $\log P(A, y|\Theta)$ under the distribution $p(A|y, \Theta^{(old)})$, where $\Theta^{(old)}$ is the hyperparameters obtained from the previous iteration. In the M-step, we maximize over Θ the expectation of the log-transformed complete likelihood conditioned on the old hyperparameters to obtain the new hyperparameters:

$$\Theta^{(new)} = \arg\max_{\Theta} E[\log P(A, y|\Theta)|y, \Theta^{(old)})]$$

So the EM algorithm works by maximizing $E[\log P(A, y|\Theta)|y, \Theta^{(old)})]$ with respect to Θ, given the previous estimate $\Theta^{(old)}$, and then replace it in the next iteration with the new Θ obtained by maximization in the previous step. The EM algorithm was

shown to increase the likelihood in each iteration and will converge to the (local) maximum [95].

The expectation $E[\log P(A, y|\Theta)|y, \Theta^{(old)})]$ can be written out as

$$\sum_A p(A|y, \Theta^{(old)}) \log p(A, y|\Theta)$$

Again, we meet with the problem of having to deal with the task of summing over all possible segmentations. To avoid this, we use a variant of EM called stochastic EM. The difference between stochastic EM and ordinary EM is that we use samples from the distribution $p(A|y, \Theta^{(old)})$ to approximate the summation, that is, we replace the summation above with samples drawn from the posterior using hyperparameters obtained from the last iteration:

$$E[\log P(A, y|\Theta)|y, \Theta^{(old)})] \approx \frac{1}{N} \sum_{n=1}^{N} \log p(A^{(n)}, y|\Theta)$$

where $\{A^{(n)}\}_1^N$ are samples from $p(A|y, \Theta^{(old)})$. But now, we can no longer guarantee that the algorithm will return a (local) maximum of the likelihood, or even that it will converge at all, although it generally works quite well in practice, and arbitrary accuracy can be obtained with a large number of samples.

The sum $\sum \log p(A^{(n)}, y|\Theta)$ can be written as

$$\begin{aligned}
\sum \log p(A^{(n)}, y|\Theta) &= \sum \log p(A^{(n)}|\Theta) p(y|A^{(n)}, \Theta) \\
&= \log \prod_{\delta,\delta'=1}^{D} K(\delta, \delta')^{n_{\delta\delta'}} + \log \prod_{\delta=1}^{D} \pi(\delta)^{n_\delta} + \\
&\quad \sum_\delta \sum_{(i,j)\in A_\delta} \log p_\delta(i - j + 1|\alpha_\delta, \beta_\delta) + \\
&\quad \sum_\delta \sum_{(i,j)\in A_\delta} \log P(y_{i:j}|\delta, 1, \Theta_\delta)
\end{aligned}$$

where $n_{\delta\delta'}$ is the count of the number of transitions from δ to δ' in $\{A^{(n)}\}_1^N$. n_δ is the

number of times $\{A^{(n)}\}_1^N$ starts in state δ, A_δ contains all segments of $\{A^{(n)}\}_1^N$ that is in state δ, and $\Theta_\delta = \{\mu_\delta^{(h)}, k_\delta^{(h)}, \nu_\delta^{(h)}, \sigma_\delta^{2(h)}\}$ is the part of hyperparameters in the conjugate prior related to state δ in the observation model.

One important observation is that in the above, the optimization can be carried out independently for each summand. The hyperparameters Θ has been partitioned into four parts. The first part consists of those in the transition matrix for the state transition probability. The second part consists of initial probabilities for each of the states. The third part consists of the shape and scale parameters of the Gamma distribution used in modeling the length of each segment. Last, the hyperparameters for the priors on the hidden parameters, the mean and variance of each segment, are separated from other hyperparameters. Moreover, the parameters for different states are separated. This is the result of conditioning on the sampled segmentations, which decouples the hyperparameters into different groups, which makes our optimization problem much easier. Actually we only have optimization problems with no more than 4 dimensions. The optimized value for parameters $K(\delta, \delta')$ and $\pi(\delta)$ is easily seen to be $n_{\delta,\delta'}/\sum_{\delta'} n_{\delta,\delta'}$ and $n_\delta/\sum_{\delta'} n_{\delta'}$ respectively, and the other parameters are optimized using IMSL function *min_col_gen_lin*. A simpler way to estimate the shape and the scale parameters are as follows. For each state δ, we collect all the segments from the samples that is in state δ, then we compute the mean m_δ and variance v_δ of the lengths. The shape parameter α_δ and scale parameter β_δ is estimated by matching the first two moments:

$$\alpha_\delta/\beta_\delta = m_\delta,\ \alpha_\delta/\beta_\delta^2 = v_\delta$$

, which leads to

$$\alpha_\delta = m_\delta^2/v_\delta,\ \beta_\delta = m_\delta/v_\delta$$

. This is simpler than doing numerical optimization, although it does not fit within our framework of maximizing the likelihood.

1.2.5 Why I do not see stochastic EM converge?

Using EB framework with stochastic EM algorithm, we can estimate our hyperparameters in a principled way. However, when implemented as above, the hyperparameters do not seem to converge to reasonable values most of the time. In particular, the hyperparameter $\sigma^{(h)}$ is observed to shrink down to 0 after several iterations. Note that there is no a priori reason that the maximum likelihood estimate should exist. It turns out this is exactly the problem. Actually, similar problems have been observed in other models before.

Modeling clustered data by a mixture model, in particular, a mixture of Gaussian (MoG) is a popular approach. In an MoG model, the data are assumed to be generated from the following mixture distribution:

$$\sum_{i=1}^{c} p_i \frac{1}{\sigma_i} \varphi(\frac{x-\mu_i}{\sigma_i})$$

where c is the number of components in the model, p_i is the mixing coefficient of the ith component with the constraint $\sum_i p_i = 1$, and $\varphi(x)$ is the standard Gaussian density function with mean 0 and variance 1. The observations are generated by first picking a component with probabilities proportional to the mixing coefficients, then drawing a sample from the normal distribution which we have picked. The parameters to be learned are $\{p_i, \mu_i, \sigma_i\}_{i=1}^{c}$. The standard method for solving this problem is the EM algorithm, where the missing data are the assignment of each observation to one component.

The EM algorithm typically converges very fast to produce a reasonable estimate of the parameters. However, this good behavior observed in practice is probably the main reason for the over-optimistic attitude towards the estimation problem. It is well-known that the likelihood for the mixture model can be arbitrarily large. In particular, suppose we have a model with two components. The observations are

$\{y_i\}$, with the likelihood

$$L = \prod_{i=1}^{n} \left(p_1 \frac{1}{\sigma_1} \varphi(\frac{y_i - \mu_1}{\sigma_1}) + p_2 \frac{1}{\sigma_2} \varphi(\frac{y_i - \mu_2}{\sigma_2}) \right)$$

We obviously have the inequality

$$L \geq p_1 \frac{1}{\sigma_1} \varphi(\frac{y_1 - \mu_1}{\sigma_1}) \prod_{i=2}^{n} p_2 \frac{1}{\sigma_2} \varphi(\frac{y_i - \mu_2}{\sigma_2})$$

by the positivity of both summands. We fix some values for $p_1 > 0, p_2 > 0, \mu_2$ and $\sigma_2 > 0$ and set $\mu_1 = y_1$. If we then let σ_1 goes to zero, $\frac{1}{\sigma_1} \varphi(\frac{y_1 - \mu_1}{\sigma_1}) = \varphi(0)/\sigma_1$ will diverge to infinity. [18] conducted extensive simulations and found that the EM algorithm usually converges to the true parameters. But once in a while, the variance parameter will converge to 0. This illustrated that the estimation in an MoG model depends on our luckiness that the parameters are actually trapped in the local maximum. The reason that this point is usually glossed over is that the convergence of variance to 0 only happens rarely, and can usually be avoided using good parameters in the initialization step.

In our hierarchical model, we observe that the problem is much more severe. Typically the parameters will fail to converge. Even when it seems to converge after a few iterations, if we continue running the program, some parameters will always eventually converge to 0. This more serious problem that occurs in our model might be due to the higher dimension we are working with, and also the fact that we used the stochastic version of EM which makes it possible that the parameters can alway escape from local maximum just by chance.

We will demonstrate that a very similar problem occurs in our model, that the likelihood can grow without bound with properly chosen hyperparameters. This will imply that some measures must be taken to properly deal with this situation. The marginal probability that is to be maximized is $p(y) = \sum_A p(y|A)p(A)$. We

construct a segmentation $A_0 = \{\delta_0, c_1, \delta_1, c_2 = n\}$ that consists of only two segments with different states $\delta_0 \neq \delta_1$. The first segment only contains the first observation. This construction is similar to what we used in the MoG model, where only one observation is generated from the first component, and all the other come from the second component. We obviously have

$$p(y) > p(y|A_0)p(A_0)$$

We will show that we can make $p(y|A_0)$ arbitrarily large by choosing appropriate hyperparameters $\{\mu_\delta^{(h)}, \sigma_\delta^{2(h)}\}$, with all other parameters fixed. This will accomplish our goal of showing the unboundedness of the marginal likelihood since $p(A_0)$ is strictly positive.

We have $p(y|A_0)p(A_0) = p(y_1|\delta_1)p(y_{2:n}|\delta_2)p(A_0)$. We presented an explicit formula for $p(y_1|\delta_1)$ in section 1.2.2. By the formula (1.3) there, with $n = 1$, we have

$$p(y_1|\delta_1) = \frac{1}{\sqrt{2\pi}}\sqrt{\frac{k_{\delta_1}^{(h)}}{k_{\delta_1}^{(h)}+1}}\frac{g(\nu_{\delta_1}^{(h)}, \sigma_{\delta_1}^{2\ (h)})}{g(\nu_{\delta_1}^{(h)}+1, \frac{\nu_{\delta_1}^{(h)}\sigma_{\delta_1}^{2\ (h)}+\lambda}{\nu_{\delta_1}^{(h)}+1})}$$

To ease notation, in the following, the sub- and super-script in the hyperparameters are omitted, $\mu_{\delta_1}^{(h)}, k_{\delta_1}^{(h)}, \nu_{\delta_1}^{(h)}, \sigma_{\delta_1}^{(h)}$ are simply written as μ, k, ν, σ.

If we choose $\mu_{\delta_1}^{(h)} = y_1$, then we get

$$\begin{aligned}
\lambda &= k\mu^2 + y_1^2 - \frac{(k\mu + y_1)^2}{k+1} \\
&= \frac{k^2\mu^2 + k\mu^2 + ky_1^2 + y_1^2 - (k^2\mu^2 + 2k\mu y_1 + y_1^2)}{k^{(h)}+1} \\
&= 0
\end{aligned}$$

So we have

$$\begin{aligned}\frac{g(\nu,\sigma^2)}{g(\nu+1,\frac{\nu\sigma^2+\lambda}{\nu+1})} &= \frac{(\frac{\nu}{2})^{\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})}\frac{\Gamma(\frac{\nu+1}{2})}{(\frac{\nu+1}{2})^{\frac{\nu+1}{2}}}\frac{\sigma^\nu}{(\frac{\nu\sigma^2}{\nu+1})^{\frac{\nu+1}{2}}} \\ &=: a(\nu)*(\sigma)^{-1}\end{aligned}$$

Now it is easy to see that by letting $\mu_{\delta_1}^{(h)} = y_1$, the marginal likelihood will diverge to infinity as $\sigma_{\delta_1}^{(h)} \to 0$, with all other hyperparameters fixed.

So the unstable behavior we observed is not a problem with our algorithm itself, it is caused by the simple fact that the maximum likelihood estimator does not exist. This is just one manifestation of the fact that mixture models are typically irregular model. To address the problem in an MoG model, [18] used penalized regression. We will follow this strategy.

With penalization, the M-step of EM will be to maximize $E[\log P(A,y|\Theta) + \log\gamma(\Theta)|A,\Theta^{(old)}]$. So the penalization is equivalent to putting a prior on the hyperparameters. But be warned that we are not using a fully Bayesian model here. The prior $\gamma(\Theta)$ is only used to estimate the hyperparameters. The inference is still done conditioned on the estimated values of the hyperparameters, which we do not integrate over.

In runs without penalization, we observed that the hyperparameters $k_\delta^{(h)}$ and $\nu_\delta^{(h)}$ usually will diverge to infinity, while $\sigma_\delta^{(h)}$ will shrink down to zero. So we use a Gamma prior on both $k_\delta^{(h)}$ and $\nu_\delta^{(h)}$, and an inverseGamma prior on $\sigma_\delta^{(h)}$. An inverseGamma density function approaches 0 near the origin, so it will pull the estimate away from it. We used the prior $Gamma(10,1)$ (with a mean of 10 and a standard deviation of 3.2) on both $k_\delta^{(h)}$ and $\nu_\delta^{(h)}$, the prior $Inv-Gamma(10,10)$ (with a mean of 1.1 and a standard deviation of 0.4) on $\sigma_\delta^{(h)}$. These parameters for the priors are chosen somewhat arbitrarily. So we have

$$\gamma(\Theta) = \prod_{\delta=1}^{D} G(k_\delta^{(h)}|10,1) \times G(\nu_\delta^{(h)}|10,1) \times Inv-G(\sigma_\delta^{2(h)}|10,1)$$

where $G(\cdot|\alpha,\beta)$ and $Inv-G(\cdot|\alpha,\beta)$ are the density function for the Gamma and inverseGamma distributions respectively.

After adding in penalization, the EM algorithm usually appears to be stable after 5 to 10 iterations. Although the stochastic EM algorithm saves computation time, the recursion still needs to be done multiple times, once for each iteration. For this training, in the application to be presented in the next section, we employed a subset of the available data, the longer ungapped sequences with at least 100 observations without intermittent missing observations.

1.2.6 Yet another stochastic EM

We implemented the EM algorithm presented above for our application in the next section. It turns out another slightly different stochastic EM algorithms can be employed. In the EM algorithm above, we consider the segmentation as our missing data, the complete likelihood is thus $p(y,A|\Theta)$, and the hidden parameters — the mean and variance for each segment are still being integrated out of the expression. But we can also treat the hidden parameters as missing variables, this leads to the complete likelihood $p(y,A,\{\mu_i,\sigma_i^2\}|\Theta)$. Another EM algorithm can be designed based on this "more complete" likelihood. By explicitly incorporating the extra hidden parameters into the likelihood, we can further simplify the optimization step.

In the following, we think of the hyperparameters as consisting of three parts:

- the hyperparameters associated with the likelihood $p(y|A)$:

  $\theta=\{\mu_\delta^{(h)},k_\delta^{(h)},\nu_\delta^{(h)},\sigma_\delta^{(h)}\}$

- the hyperparameters associated with the length distribution: $l=\{\alpha_\delta,\beta_\delta\}$

- the hyperparameters associated with the Markov chain transitions:

  $\tau=\{K(\delta,\delta'),\pi(\delta)\}$

The EM algorithm is similar to the one used before, and the expectation of the complete likelihood is:

$$E[\log p(y, A, \{\mu_i, \sigma_i^2\}|\theta, \tau, l)|y, \theta^{(old)}, \tau^{(old)}, l^{(old)}]$$

The complete likelihood can be written as the sum of three terms:

$$\log p(y, A, \{\mu_i, \sigma_i^2\}|\theta, \tau, l) = \log p(A|\tau, l) + \log p(\{\mu_i, \sigma_i^2\}|A, \theta) + \log P(y|A, \{\mu_i, \sigma_i^2\})$$

Note that the last term does not contain any hyperparameters.

The EM algorithm consists of iterations of two steps:

- (stochastic)E-step:

  draw N samples $\{A^{(n)}, \{\mu_i^{(n)}, \sigma_i^{2(n)}\}\}_{n=1}^N$ from the posterior conditioned on the hyperparameters from the previous iteration:

  $$A, \{\mu_i, \sigma_i^2\} \sim p(A, \{\mu_i, \sigma_i^2\}|y, \theta^{(old)}, \tau^{(old)}, l^{(old)})$$

- M-step:

  optimize $\sum_{n=1}^N \log p(A^{(n)}|\tau, l)$ to get the new hyperparameters τ, l.

  optimize $\sum_{n=1}^N \log p(\mu_i^{(n)}, \sigma_i^{2(n)}|A^{(n)}, \theta) + \log \gamma(\theta)$ to get the new hyperparameters θ.

In the above we use the superscript (n) to indicate different samples, the subscript i to index the segments contained in $A^{(n)}$. Since

$$p(A, \{\mu_i, \sigma_i^2\}|y, \theta^{(old)}, \tau^{(old)}, l^{(old)}) = p(A|y, \theta^{(old)}, \tau^{(old)}, l^{(old)}) \cdot p(\{\mu_i, \sigma_i^2\}|y, A, \theta^{(old)})$$

, to draw samples in the E-step, we can first draw A using backward sampling. Given A, the hidden parameters for each segment can be drawn from the posterior

conditioned on A.

$$\mu_i, \sigma_i \sim N - inv\chi^2(\mu_n, \frac{\sigma_n^2}{k_n}; \nu_n, \sigma_n^2)$$

where the formula for updating the parameters in the $Normal - inverse - \chi^2$ distribution appeared in section 1.2.2.

The optimization for τ, l is same as before, by counting the number of transitions between segments in the samples for each pair of states, and using the lengths of segments in each state. In order to optimize over θ, we split the hidden parameter $\{\mu_i^{(n)}, \sigma_i^{(n)}\}$ according to the states of the segments into D groups, and the optimization is done separately for each group. Fix one group $\{\mu_i, \sigma_i^2\}_{i=1}^m$ for a particular state δ, say, we show that some of the hyperparameters can be optimized analytically.

We want to maximize $\prod_i p(\mu_i, \sigma_i^2|\delta, \theta)\gamma(\theta)$ over θ. The objective function can be written out as (omitting the subscripts and superscripts in the hyperparameters)

$$\frac{1}{\prod_{i=1}^n \sqrt{2\pi\sigma_i^2/k}} e^{-k\sum_i \frac{(\mu_i-\mu)^2}{2\sigma_i^2}} \prod_i \frac{(\frac{\nu}{2})^{\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})} \sigma^\nu (\sigma_i^2)^{-(\frac{\nu}{2}+1)} e^{-\frac{\nu\sigma^2}{2\sigma_i^2}}.$$

$$\frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} k^{\alpha_k - 1} e^{-\beta_k k} \frac{\beta_\nu^{\alpha_\nu}}{\Gamma(\alpha_\nu)} \nu^{\alpha_\nu - 1} e^{-\beta_\nu \nu} \frac{\beta_\sigma^{\alpha_\sigma}}{\Gamma(\alpha_\sigma)} (\sigma^2)^{-\alpha_k - 1} e^{-\beta_\sigma/\sigma^2}$$

where those α, β's with subscripts are the parameters for the Gamma or inverseGamma density used in the penalization term $\gamma(\theta)$.

First, the optimal μ can be found by minimizing

$$\sum_i \frac{(\mu_i - \mu)^2}{2\sigma_i^2} \tag{1.11}$$

which gives us $\mu = (\sum_i \frac{1}{\sigma_i^2)}^{-1} \sum_i \frac{\mu_i}{\sigma_i^2}$. Denoting the minimized value in (1.11) by r, after plugging in the solution for μ, we see that the optimal k should maximize

$$\sqrt{k}^m e^{-kr} k^{\alpha_k - 1} e^{-\beta_k k}$$

which gives us $k = \frac{m/2+\alpha_k-1}{r+\beta_k}$. The other two parameters ν, σ^2 is solved by numerical method. Note that with the first stochastic EM in section 1.2.4, we need to solve a four dimensional problem numerically.

1.2.7 Missing observations

Our model also yields a principled means to span the missing observations that goes beyond the implicit geometric length distributions of HMMs. When the gaps are small, we expect the states at the beginning and end of the gaps to be the same, but this correlation across missing observations is unlikely to span long gaps. Our model describing the lengths of substrings and state transitions yields inferences on the state of the observation at the end of a gap as a function of the gap length. The inference of this missing data borrows information from the other side of the gap. Over a gap, our Gamma distribution model of substring length infers the number of change points in the gap, and the state transition model determines the distribution of states after several state transitions in the gap.

Let's suppose that on a subset $I \subseteq \{i, i+1, \ldots, j\}$ we make no observations. In the forward recursion, we need to compute the probability $p(y_{obs}|\delta, 1)$, where y_{obs} is the observed values, which can be written as the integral over the missing observations

$$p(y_{obs}|\delta, 1) = \int p(y_{i:j}|\delta, 1) dy_{mis}$$

We can simply ignore the missing observations in the computation of $p(y_{i:j}|\delta, 1)$, the missing observation is taken into account only when we compute the length distribution. This sounds more complicated than it really is. In fact the recursion and sampling can be done exactly as before by setting $p(y_{j:j'}|\delta, 1) = 1$ if all values inside $[j, j']$ are missing.

We point out that in fact we made an important assumption above which may

not be obvious to the reader. Without any assumption, we have

$$p(y_{obs}, \mathbf{r}) = \int p(y_{obs}, y_{mis}|\mu_i, \sigma_i^2)p(\mathbf{r}|y_{mis}, y_{obs}, \phi)p(\mu_i, \sigma_i^2|\Theta)d\mu_i\sigma_i^2 dy_{mis}$$

where $\mathbf{r}$ is an indicator vector containing the positions of the missing observations, and ϕ contains all other parameter that $\mathbf{r}$ can depend on. Because of the integration over y_{mis}, this will potentially introduce complicated correlations between components of y_{obs}. So the integration over μ and σ generally cannot be computed. If we instead assume that $p(\mathbf{r}|y_{mis}, y_{obs}, \phi) = p(\mathbf{r}|y_{obs})$, that is all the information contained in $\mathbf{r}$ can be derived from the observed data, then it can be easily seen that integration over y_{mis} will not introduce any correlation between the observed data give μ_i and σ_i^2, so $p(y_{obs})$ can be computed by ignoring the missing data. This assumption is called Missing At Random (MAR)[52]. This is the working assumption under which our approach works.

1.2.8 Numerics

Direct implementation of our algorithm only works for sequences of a couple hundred observations long. For longer sequences, the underflowing or overflowing of float numbers should be properly dealt with. In computing the MAP estimate, this problem is easily solved by using log probability and thus transform the products into the sums. This does not solve our problem in forward recursion and some other parts of the algorithm because sums are mixed together with products in the recursion.

In the following we explain our approach to solve this problem. The following explanations are also of independent interest, and we illustrate it with the computation of $p(i, \delta, k)$. The main idea is actually the same as in computing the MAP. In the computer memory, the probability $p(i, \delta, k)$ is stored by its log $lp(i, \delta, k) := \log p(i, \delta, k)$. Computation of $\log p(i, \delta, k)$ by recursion (1.4) should proceed with care. We have

the recursion:

$$p(i,\delta,k) = \sum_{j<i,\delta'} p(j,\delta',k-1)p_\delta(i-j)K(\delta',\delta)$$

Suppose the log of $p(j,\delta',k-1)$ in the sum has been stored in the memory, we might be tempted to write everything in the exponential from, in which case we have

$$p(i,\delta,k) = \sum_{j,\delta'} e^{lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)}$$

But taking exponential directly does not work — the reason that we choose to store lp instead of $p = e^{lp}$ in the first place is because computation of e^{lp} might lead to overflow or underflow. This is still true with $e^{lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)}$. Here is the trick to avoid this. Let $c = \max_{j,\delta'}\{lp(j,\delta',k-1) + \log p_\delta(i-j) + \log K(\delta',\delta)\}$, so

$$\begin{aligned} p(i,\delta,k) &= \sum_{j,\delta'} e^{lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)} \\ &= (\sum_{j,\delta'} e^{lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)-c})e^c \end{aligned}$$

Now the numbers in the exponent $lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)-c$ is non-positive, with the largest among them exactly 0. Although after taking exponential, their might be some positive terms that will become 0 due to underflow, we can be sure that these terms do not have significant contributions in the summation, because the sum is dominated by terms that are close to 1. In this way we separate the terms that are the dominating ones from the others, and these terms are computed with high precision. The probability $p(i,\delta,k)$ will finally be stored in memory as

$$\log(\sum_{j,\delta'} e^{lp(j,\delta',k-1)+\log p_\delta(i-j)+\log K(\delta',\delta)-c}) + c$$

1.2.9 Model selection

In many applications, the number of states used in an HMM can often be predetermined. For example, in part-of-speech tagging, the number of states can be the same as the number of possible tags. In this problem, it is quite obvious how many states one should use. In our application in the next section, there does not exist clear-cut biological theory that tells us how many states there are, so it would be better if the number of states can be inferred from the data.

One popular model selection method is the AIC [1]. When maximum likelihood is used for inference, a natural criterion for the "goodness-of-fit" is the value of maximized likelihood. However, when we compare several nested models using maximum likelihood, the more complex model will always have higher likelihood, since the likelihood is being optimized on a larger space. So it cannot be used to select the model. What we actually want is the expected value of the log likelihood $E(\log p(y|\hat{\theta}_{ML}))$, where the expectation is taken over observation y, conditioned on the estimated parameter. The larger the expected likelihood, the closer it is to the true model in the Kullback-Leibler distance. Since the true model p_0 is unknown, we must estimate it using the data. The natural approach is to replace the expectation by the sample average $\frac{1}{n}\sum \log p(y_i|\hat{\theta}_{ML})$. The problem of using the sample mean as an estimate of $E(\log p(y|\hat{\theta}_{ML}))$ is that the same data that was used to estimate the parameter is used to approximate the expectation, which produces bias in our estimate. Using Taylor expansion, Akaike[1] approximated the bias using the dimension of the parameter, p , i.e. the model that obtains the highest value of $\sum \log p(y_i|\hat{\theta}_{ML}) - p$ is selected. There are various versions of AIC with slightly different penalty terms in different contexts, all trying to get a more accurate approximation for the bias, based on the same principle as described above. To use AIC, the objective function does not have to be the likelihood. General loss function L can be used. The new objective is to minimized $E(L(y;\theta))$. The maximum likelihood is just one particular case with negative log loss $L = -\log p(y|\theta)$. In particular, AIC can be used within the penal-

ized maximum likelihood framework [59]. In our hierarchical model, for example, the number of parameters for a 3-state model would be 26 (12 parameters in the prior for hidden parameters, 8 for transition matrix and initial distribution, 6 in the Gamma distribution for the length of segments).

Another widely used information criterion is BIC[76]. This is motivated by Bayesian model selection, where the model with the largest posterior probability $p(M|y)$ is selected. By the Bayes rule $p(M|y) = \frac{p(y|M)p(M)}{p(y)}$, which is proportional to $p(y|M)$ if different models are assigned uniform probability a priori. The best model is selected based on the marginal likelihood given the model, after the parameter is integrated out. Formally, the Bayesian model selection works by computing the integral $\int p(y|\theta, M)p(\theta|M)d\theta$. The relative likelihood between two competing models $\frac{p(y|M_1)}{p(y|M_2)}$ is called Bayes Factor. Using Laplace approximation around the maximum likelihood estimate, the integral can be approximated which leads to BIC, or Bayesian Information Criterion. The surprising fact is that BIC does not depend on the prior chosen. Although BIC is derived under a Bayesian framework, the derived criterion is used in a frequentist sense. Formally, BIC chooses the model that maximizes $\sum \log p(y_i|\hat{\theta}_{ML}) - \frac{1}{2}p \log n$, where n is the number of independent observations. In some simple cases, the AIC and BIC have both been shown to lead to consistent model selection criterion[77][78], but under different kind of asymptotics. Although derived from fundamentally different principles, BIC bears a striking similarity to AIC in form. The only difference between AIC and BIC is the extra factor $\log n/2$ in penalty term per parameter, so BIC will generally select a simpler model compared to AIC. By convention, AIC and BIC are actually defined as the negative log likelihood plus the penalty term, i.e. $-\sum \log p(y_i|\hat{\theta}_{ML}) + p$ and $\sum \log p(y_i|\hat{\theta}_{ML}) - \frac{1}{2}p \log n$ respectively.

AIC can be directly used in our problem. But application of BIC is a little bit tricky. $\log n$ is used in the penalty term of BIC. Classical BIC deals with problems with observations that are i.i.d., and n is the number of observations. Since our

observations are not independently drawn from the hyperprior $p(\Theta)$, using sequence length as n is not appropriate. We discuss what we think might be the appropriate penalization to use next. Since each segment has its own hidden parameters, roughly speaking each segment with its length and assigned state provides us with only one observation from the hyperprior. So setting n to be the number of segments seems to be more appropriate, although we have no rigorous justification for its use. In practice, the number of segments can be estimated from multiple samples of the segmentation.

For our application in the next section, we actually do not need to use either of the criteria above. The sequence data consist of about 400,000 observations. We only use about 15% of those in our training to estimate the hyperparameters. Using all data for training is infeasible since the EM algorithm requires us to scan through the data multiple times. This leaves us with the possibility of using the rest of the data that is not involved in training as a independent validation set to select appropriate models. So we can estimate $E(\log p(y|\hat{\theta}_{ML}))$ using the sample average unbiasedly, avoiding the problem mentioned above.

1.2.10 Connections with GHMM

In this subsection, we describe the Generalized Hidden Markov Model (GHMM) to see what is its connections and differences with our model. In particular, we compare the unconventional forward recursion used in our model to that in classical GHMM.

One obvious and probably the most important difference is that we used a hierarchical approach where each state is actually a continuous mixture. This makes our model much more flexible than GHMM. Also because of this additional level, the usual EM for GHMM becomes difficult to apply, this is the reason that we use the stochastic version in our estimation procedure. and the usual backward recursion is replaced by our backward sampling. Other than these differences, we will see these two models are actually almost identical.

We use the commonly adopted α, β notation for GHMM. Let

$$\begin{aligned}
\alpha_i(\delta) &= p(y_{1:i}, \text{ with last state } \delta) \\
\beta_i(\delta) &= p(y_{i+1:n} | \text{ with state ending at } i \text{ being } \delta)
\end{aligned}$$

We have the following recursion:

$$\begin{aligned}
\alpha_0(\delta) &= \pi(\delta) \\
\alpha_i(\delta) &= \sum_{j<i,\delta'} \alpha_j(\delta') K(\delta', \delta) p_\delta(i-j) p(y_{j+1:i}|\delta, 1) \\
\beta_n(\delta) &= 1 \\
\beta_i(\delta) &= \sum_{j>i,\delta'} K(\delta, \delta') p_{\delta'}(j-i) p(y_{i+1:j}|\delta', 1) \beta_j(\delta')
\end{aligned}$$

The recursion above is very similar to the forward recursion (1.5). In fact, as we point out, they are approximately equivalent to one another, but not exactly.

If we multiply both sides in (1.5) by $p(i, \delta, k)$ and then sum over k. We get

$$\begin{aligned}
& \sum_k p(i, \delta, k) p(y_{1:i}|\delta, k) \\
= & \sum_k \sum_{j<i,\delta'} p(j, \delta', k-1) p_\delta(i-j) K(\delta', \delta) p(y_{0:j}|\delta', k-1) p(y_{j+1:i}|\delta, 1) \\
= & \sum_{j<i,\delta'} \left(\sum_k p(j, \delta', k-1) p(y_{0:j}|\delta', k-1) \right) p_\delta(i-j) K(\delta', \delta) P(y_{j+1:i}|\delta, 1)
\end{aligned}$$

In a GHMM, the number of segments k could be as big as the number of observations. If we sum k from 1 up to n, we see that $\sum_k p(i, \delta, k) p(y_{1:i}|\delta, k) = p(y_{1:i}, \delta)$, which is exactly $\alpha_i(\delta)$. In our recursion in section 1.2.2, we only perform the recursion up to k_{max}, so $\sum_{k=1}^{k_{max}} p(i, \delta, k) p(y_{1:i}|\delta, k)$ is only approximately equal to $\alpha_i(\delta)$, the difference comes from that we essentially set $p(y_{1:i}|\delta, k)$ with k larger than k_{max} to be zero.

In our model with explicit conditioning on k, the backward recursion can also be performed

$$p(y_{i+1:n}|k,\delta) = \sum_{j>i,\delta'} K(\delta,\delta')p_{\delta'}(j-i)p(y_{i+1:j}|\delta',1)p(y_{j:n}|k+1,\delta')$$

This is exactly the same kind of recursion used for β in GHMM, with additional conditioning on k. If there is no upper bound on k, we would expect $p(y_{i:n}|k,\delta)$ to be independent of k. But because of the upper bound we have, bigger k closer to k_{max} would put severe constraint on y since there can be only be $k_{max}-k$ possible segments inside $y_{i+1:n}$. If we choose k_{max} large enough, we have that $p(y_{i:n}|k,\delta) \approx \beta_i(\delta)$ for k not too close to k_{max}. For a sequence that we have reasons to believe that $k << n$, we can be confident that these two will be very close.

The advantage of performing the recursion explicitly conditioned on k is that we get exact distribution for the number of change points, while in GHMM, the information of k is not easy to recover since we have implicitly summed over it. In GHMM, an approximate posterior distribution on the number of change points can be obtained from samples, which are possible to obtain after some extra work.

1.2.11 Analysis of computational complexity

A simple implementation of recursion (1.5) actually takes time cubic in the length of the sequence, because the computation of $p(y_{i:j}|\delta,1)$ takes $O(i-j)$ time. A careful analysis shows that the computational complexity is $O(k_{max}D^2n^3)$, where D is the number of states, and its memory requirement is $O(k_{max}Dn)$ (used to store $p(y_{1:i}|k,\delta)$). We will show that by carefully arranging the computations involved, we can get by with $O(k_{max}D^2n^2)$ computation time without increasing the memory required. In some applications, when the number of states D is large, (for example, when we need to select the correct number of states, we need to perform the recursion with bigger D. The recursion above is not practical for such large state space with a

long sequence. We show that the computational complexity can easily be reduced to $O(k_{max}(Dn^2 + D^2n))$.

From section 1.2.2, we get the formula $p(y_{i+1:i+l}|\delta, 1) = \frac{1}{(2\pi)^{\frac{l}{2}}}\sqrt{\frac{k_\delta^{(h)}}{k_\delta^{(h)}+l}}\frac{g(\nu_\delta^{(h)},\sigma_\delta^{2(h)})}{g(\nu_\delta^{(h)}+l,\frac{\nu_\delta^{(h)}\sigma_\delta^{2(h)}+\lambda}{\nu_\delta^{(h)}+l})}$. Computation of λ takes linear time in the length l, while all other parts can be computed in $O(1)$. Note we have

$$\lambda = k_\delta\mu_\delta^2 + l\bar{y}^2 - \frac{(k_\delta\mu_\delta + \sum_i y_i)^2}{k_\delta + l} + \sum(y_i - \bar{y})^2$$

which is a function of $\sum y_i$, and $\sum y_i^2$. From this expression, it is obviously that the λ for $y_{j-1:i}$ can be computed in $O(1)$ given λ for $y_{j:i}$ — all we have to do is just remember $\sum y_i$ and $\sum y_i^2$ from the previous step, and update it with y_{j-1}. So, if for fixed (k, δ, i, δ'), the sum in recursion (1.5) over j is computed starting from $j = i - 1$ and decreasing to 1, the term $p(y_{j:i}|\delta, 1)$ can be computed in $O(1)$. So the total recursion takes time $O(k_{max}D^2n^2)$.

So both the recursion for $p(i, \delta, k)$ and that for $p(y_{1:i}|\delta, k)$ takes $O(k_{max}D^2n^2)$ time to compute. Actually, a more careful analysis shows that it is possible to reduce the computation time for both recursions to $O(k_{max}(D^2n + Dn^2))$. For example, we can split recursion (1.5) into two steps:

$$\begin{aligned} p(y_{1:i}|\delta, k) &= \frac{1}{p(i, \delta, k)}\sum_{j<i} a(j, \delta, k-1)p_\delta(i-j)p(y_{j+1:i}|\delta, 1) \\ a(i, \delta, k) &= \sum_{\delta'} p(i, \delta', k)K(\delta', \delta)p(y_{1:i}|\delta', k) \end{aligned}$$

The recursion for $p(i, \delta, k)$ can be performed similarly. Also, notice that computation of $p(i, \delta, k)$ does not depend on the observed data, so the recursion and be done off-line and stored in advance.

1.2.12 Summary of the basic model

In the above, we presented a hierarchical Bayesian model that is similar to the traditional HMM. Instead of assigning one mean and one variance to each state as in HMM with Gaussian emission distribution, we assume that the mean and variance is itself a sample from another state-specific hyperprior. This simple extension in concept can go a long way in modeling many types of sequence data. Although this model is motivated by the biological application which we will present in the next section, we would prefer to consider it as a general modeling tool that has a moderate extra complexity compared to that of HMM, yet much more powerful to handle real-world data. We will see some evidence of the poor fit of HMM for our application which is rescued by our model.

The Gaussian assumption is not critical, we can use any distribution as long as the explicit integral in (1.6) can be computed, which almost always requires that some conjugacy properties be explored. Our new recursion has not only the advantage of providing exact information on the underlying change point process, but also provides a control mechanism that bounds the number of change points we consider, which provides a safeguard against overfitting. We provide a principled framework that can both estimate the hyperparameters using the data, and select a appropriate model automatically. The only free parameter in our model is k_{max}, which can easily be set to a large enough value (but still much smaller than n) so that we can have confidence that it will be sufficient for modeling the data. Further assurance can be obtained by the output of the posterior, $p(k|y_{1:n})$. Rapid decrease of this probability for large k will make us comfortable with using the bound k_{max}. The only place that makes us a little uncomfortable is the use of penalized likelihood. Using penalization is essential for our model to work by avoiding the unbounded likelihood, but it makes the estimate somewhat biased. It is unclear to us how to address this issue.

1.3 Application to Biological Data

Expression and functionality of the genetic code critically depends on the level of accessibility of DNA segments to protein. Active regulatory elements across the genome are largely contained in parts of sequences that has high affinity to interaction with potential binding proteins. Finding sites with high accessibility will be valuable for genome research since these are usually the hotspots for genetic activity, and new functional elements can be discovered by focusing on those regions. One frequently applied test of accessibility is treating the DNA sample with DNase I and measure the segments sensitivity to its attack. Those sites usually consist of many segments free of nucleosome, with reduced Histone H1 presence and acetylated Histone H3 and H4. Full genome distribution of these hypersensitive sites is difficult to obtain and previous studies usually focus on limited regions around certain genes of interest. Besides the positional mapping of the sites, the composition of segments with different levels of sensitivity is an outstanding question. Other biologically interesting questions include the mutation rate within the sites and the typical span and clustering pattern of such regions.

Some recent studies have demonstrated that cloning DNase I HSs is feasible and can lead to high-resolution map of human chromatin accessibility in vivo. Here we describe a recent study which uses high-density genomic tiling DNA microarrays that covers the ENCODE regions (about 1% of the human genome).

In this section, we apply our basic model to the biological sequence data that we have which returns global organization of the hypersensitive sites. First we give a brief introduction to the biological problem and the sequence data. In the second subsection, these regions are delineated using our HMM-like model with ordered states representing different levels of sensitivity. We present prediction accuracy of our algorithm using some validated sites. Also, we will compare our model with HMM and see some evidence that HMM fits the data poorly while our model largely

addresses this problem.

1.3.1 Biological background

Eukaryotic DNA exists in cells in a compact form. Electrostatic attraction between negatively charged DNA phosphate group and positively charged side chain of certain proteins make DNA tightly bound to the protein. The complexes they form are called *chromatin*. The most common protein in the chromatin bound to the DNA is the histone. There are five different types: H1, H2A, H2B, H3, and H4. DNA can be tightly bound to all those types except H1. Under electron micrograph, the DNA looks like a string with many beads. Each bead is called a *nucleosome*, which is the fundamental subunit in the structure. A nucleosome consists of a core of eight histone molecules wrapped with two turns of DNA. The core includes two of each type of histone except H1, which forms a strong bond with the DNA. The part of DNA that is wrapped around the protein is about 150 bp long, and the part in between the "beads" that is not in contact with the octamer, usually about 50 bp long, is called *linker DNA*. The linker regions are complexed to H1 histone and also some other proteins. It is relatively easy to remove H1 from the chromatin, while much difficult to break up the binding between DNA and other types of histones. Histones can be modified by many types of chemical processes, which can change the positions of the nucleosomes, so this structure is dynamic, and the general view is that the existence of nucleosome has a repressive effect on transcription. The octamer that binds to the core DNA will prevent it from binding with RNA polymerase or other proteins, and only the linker DNA is exposed during transcription. The binding of DNA to protein is also related to DNA bending and flexibility, although the mechanism is not fully understood.

We will refer to the part of DNA that is tightly bound to histone octamer and thus is not accessible during transcription as in the nonsensitive state, while the region that is not in contact with proteins and thus most exposed is referred to as

the hypersensitive (HS) region. The regions around the hypersensitive sites usually have intermediate sensitivity.

Nucleosome positioning refers to the organization of nucleosomes along DNA determined by local DNA structure. The positioning also depends on some of the protein characteristics which can be changed under different environment. The benefits of being able to predict nucleosome positioning include several areas in genomics. For example, the abundance of nucleosomes will generally aid in the task of gene prediction, since the genes are regulated by the nucleosomes in their vicinity. Also, some chromatin diseases that occurred are believed to be related to abnormal positioning of nucleosomes.

In our experiment, after treating the sample with DNase I, tiled-arrays that contain the 44 ENCODE regions covering about 1% of the human genome is used to map the DNA fragments. We used another set of microarrays for the hybridization of another DNA sample as control. The sequence data we get is the log transformation of the ratio of the dye intensity in two arrays.

Hidden Markov models offer an appealing modeling tool for the analysis of dye intensities from tiled arrays. These seem particularly appropriate in the analysis of data that yields inferences across a spectrum of functions. Applications of multi-state HMMs to tiled array data with each state described by a Gaussian model have already shown themselves to be very useful [50]. However, inhomogeneity in the dye intensities of probes can lead to poor fits of these models. An important consequence of poorly fitting models is that observations toward the tails of the intensity distributions often become outliers from the poorly fitting model. Such outliers have an adverse impact on state predictions of HMMs in two important ways: 1) The outliers from one state can be incorrectly predicted to be members of adjacent states. 2) For many distributions, including the Gaussian, outlying observations have an unduly large impact on parameter estimates characterizing individual states. Probe specific differences in hybridization energies are important sources of heterogeneity. Li et al.

(2005)[50] show that when sufficient array data are available, probe-specific models of intensities provide a useful means to address probe-to-probe variation.

We will use the alternate procedure developed in Section 1.2 to address these inhomogeneities that can be applied even when no replicates are available. This model seeks to capitalize on the premise that sonicated fragments that span multiple probes induce similarities in intensities across substrings of tiled array probes. This change point model assumes that within-state sequences are not homogeneous, but instead are characterized by substrings of unknown lengths each of which is homogeneous (refer back to Fig 1.1).

1.3.2 Results

We first estimate the hyperparameters using only part of the sequence that is at least 100 long without intermittent gaps, those data account for about 15% of our total observations. We use Gamma distributions shifted to the right by 4 for the prior distribution of the length of the segment. This choice leads to better separation of states in our model than using usual Gamma distributions. A possible biological justification is that a length of 4 probes for our tiled-array data roughly corresponds to two complete turns of DNA sequence around the octamer, so a lower bound of 4 is reasonable. We tried using models with one single state up to five states. The model is selected by comparing the likelihood on the rest of the data conditioned on the hyperparameters learned, as explained in section 1.2.9. Figure 1.3 plotted the marginal likelihood for our model when the number of states runs from 1 through 5. It clearly indicates that a 4-state model is appropriate. After we found that our method suggests that we have four states, the biologists we worked with decide that these states will roughly correspond to nonsensitive regions, regions with intermediate sensitivity, minor hypersensitive sites, and major hypersensitive sites, respectively. Figure 1.4 shows the AIC and BIC curve, both suggest a 4-state model.

With each probe 50 bp long and spaced at 38 bp on the microarray, the sequence

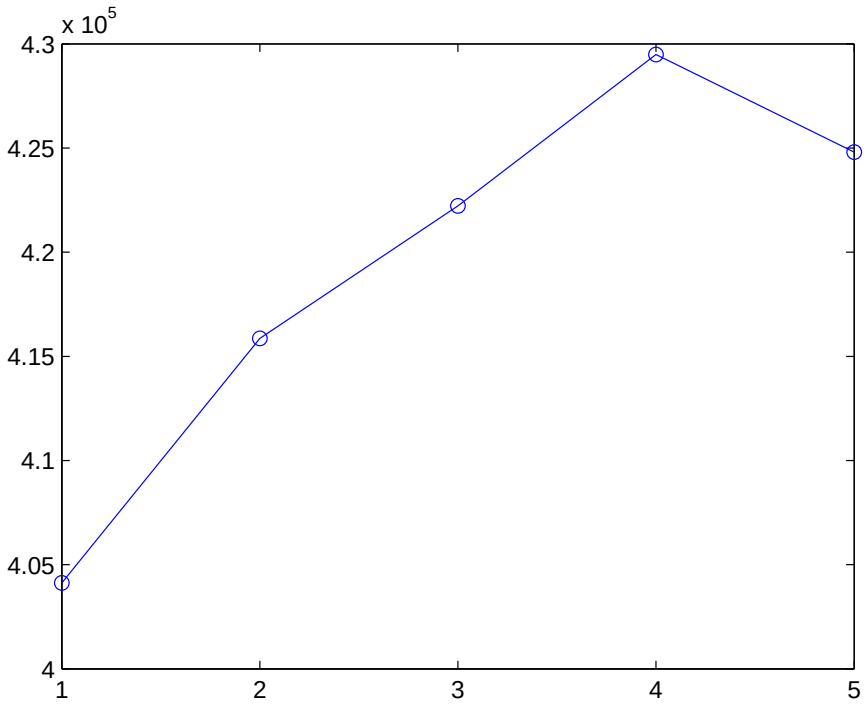


Figure 1.3: Log likelihood computed with training data excluded, conditioned on the estimated hyperparameters.

we have contains about 400,000 observations. It is impossible to apply our algorithm directly on such a long sequence, so we break up the sequence into pieces each with about 1000 probes. The hyperparameters learned are listed in Table 1.1, where it can be seen that the two states corresponding to hypersensitive sites have positive hyperprior means. An interesting observation is that the transition probability from the nonsensitive sites to the hypersensitive sites is close to zero, indicating hypersensitive sites are flanked by regions with intermediate sensitivity. Similarly, the major hypersensitive sites are typically seen within minor hypersensitive regions. In the implementation, we set k_{max} to be 100, while the posterior probability on k almost always has a maximum below $k = 10$. EM algorithm seems to become stable after only about 5 iterations. We use 500 samples to estimate the centroid and compute the uncertainty.

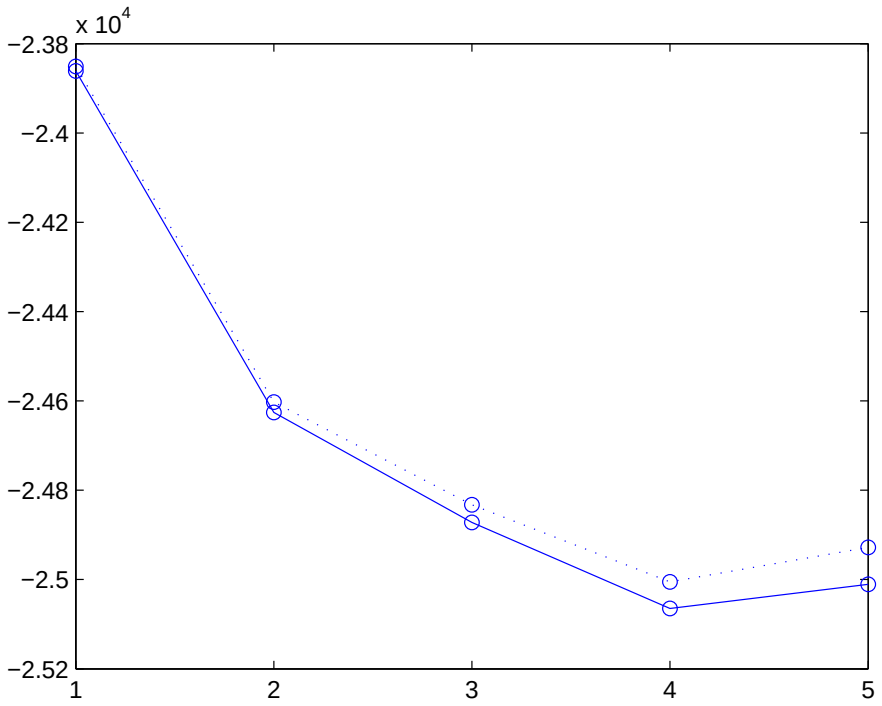


Figure 1.4: AIC (solid) and BIC (dotted) both suggest a 4-state model.

Overall, we find that there are long regions that are predicted to be entirely in the nonsensitive and intermediate states, and other regions with clusters of major and minor hypersensitive states. Figure 1.5 shows predictions from the hierarchical model for a region in chromosome 7. As mentioned before, sampling from the posterior distribution of the segmentations allows us to assess the uncertainty. The bottom panel of the figure shows the probability of the inferred state based on a sample of 500 solutions. These give an assessment of the certainty of the inferred states. The MAP estimate (not shown in the figure) looks very similar to the centroid, although we have not looked into the detail yet.

Table 1.2 lists some simple statistics of the segmentations. In computing lengths, each state is bound only by its lower-mean neighbors. I.e., an intermediate sensitive domain is bound on either side by nonsensitive states, and minor HSs are bound on either side by either nonsensitive or intermediate sensitive regions. For example, say

	$\mu_\delta^{(h)}$	$\delta'=1$	$\delta'=2$	$\delta'=3$	$\delta'=4$	α_δ	β_δ
$\delta=1$	-0.64	0.34	0.66	1e-6	1e-6	1.9	0.03
$\delta=2$	-0.35	0.67	1e-6	0.33	1e-6	1.0	0.04
$\delta=3$	0.25	0.08	0.45	1e-6	0.47	1.4	0.1
$\delta=4$	1.36	1e-6	1e-6	0.999	1e-6	0.8	0.08

Table 1.1: Some hyperparameters estimated for a 4-state model, including hyper-mean, transition probability, and the segment length parameters. For the transition probability, the number 1e-6 appears multiple times because this is the minimum transition probability allowed in our implementation.

	nonsensitive	intermediate	minor HS	major HS
number of segments	1,062	1,085	521	492
number of bp	28,007kb	1772kb	518kb	162kb
mean length	26,371	1,633	994	329
median length	13,718	950	646	228

Table 1.2: Some descriptive statistics.

we have 1000 bp in intermediate state, then 100 bp in minor hypersensitive, then another 1000 bp in intermediate state, the duration of the intermediate state will be 2100 bp. From the table, the non-hypersensitive domains occupy most of the genome. The major hypersensitive sites usually occupy a few hundred bp, which fit well with biologists' prior knowledge. Note that the length statistics is not prespecified as an input, rather they are learned from the data.

In section 1.2.2, we showed that after we obtained a sampled segmentation by backward sampling, we can calculate the posterior of the hidden parameters μ and σ^2 for each segment, which has a $normal-inverse-\chi^2$ distribution. After multiple samples are drawn, the posterior of the hidden parameters can be represented as a mixture of $normal-inverse-\chi^2$ distributions. We use the mean of this mixture distribution as our estimate of the hidden parameters μ and σ^2. The mean of μ was plotted in Figure 1.2.2 together with the raw data. It can be seen that the posterior mean of μ looks like the smoothed version of the data. Using these estimate for μ

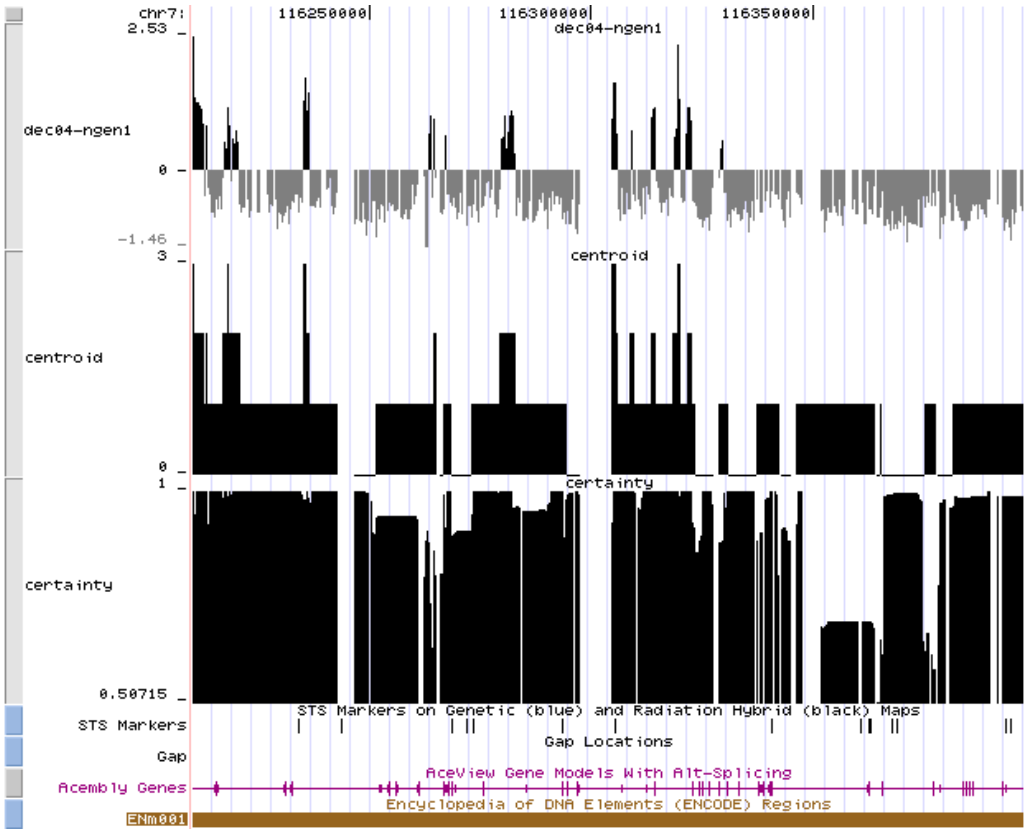


Figure 1.5: Prediction of our model shown in the UCSC genome browser. The top panel is the raw data, the middle is the centroid, the bottom is the uncertainty.

and σ^2, we can define the (standardized) residual for each observation y_i as

$$r_i = \frac{y_i - \mu_i}{\sigma_i}$$

According to our model, r_i should roughly follow a standard normal distribution. In Figure 1.6, we show the QQ-plots of the residuals against the normal distribution, which are split into four groups according to the centroid state. In Figure 1.7, we show the QQ-plots of the residuals for a 4-state HMM. The most apparent departure from normal appears in the nonsensitive state and the major HS state. QQ-plots in our model is much closer to a straight line, which shows the self-consistency of our model. We only have a small number of hypersensitive sites validated by the southern

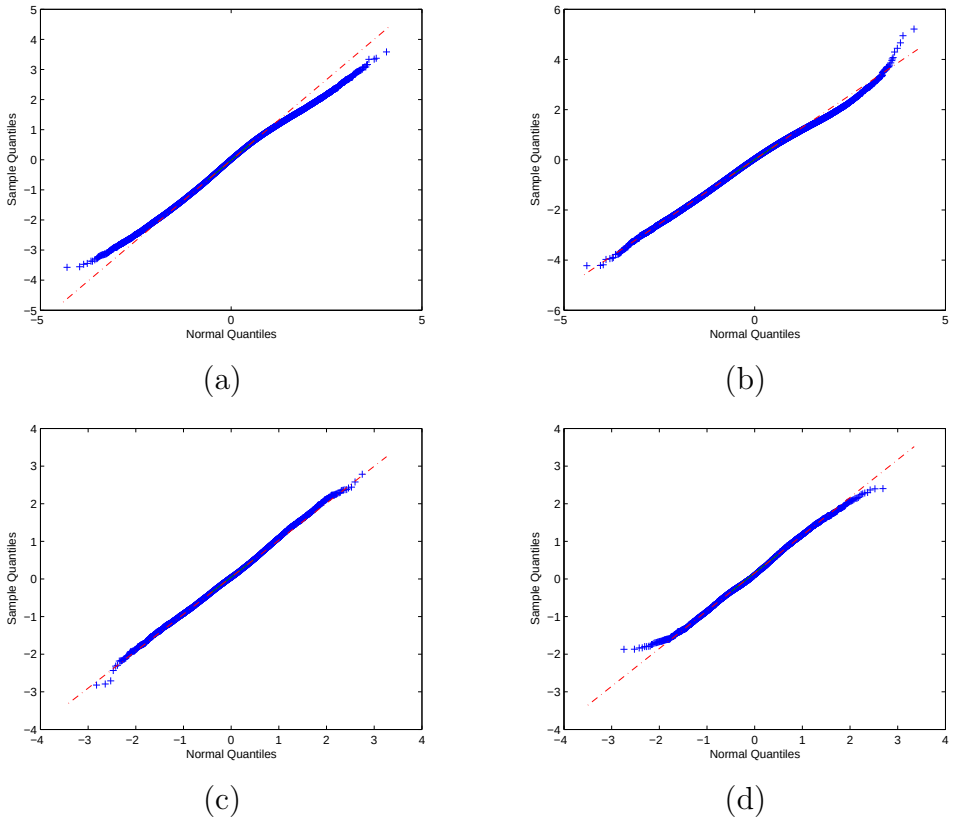


Figure 1.6: QQ-plots for the residual vs. Normal of our hierarchical model for (a) data in nonsensitive region (b) data in intermediate sensitive region (c) data in minor HS (d) data in major HS.

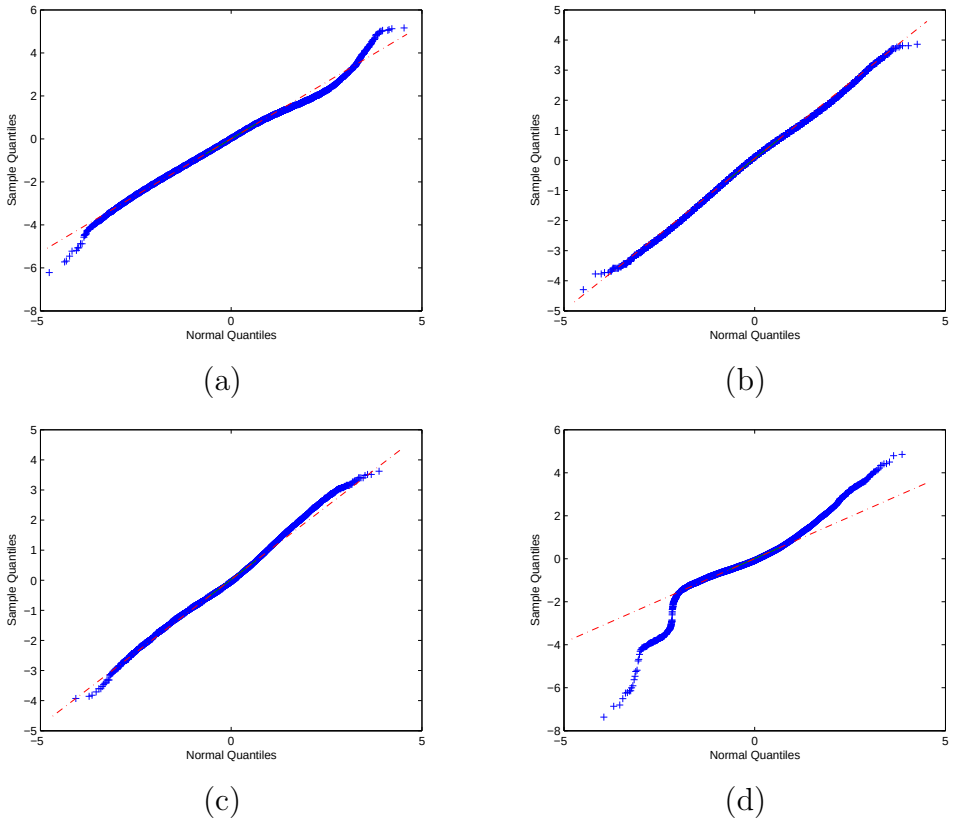


Figure 1.7: QQ-plots for the residual vs. Normal of the standard 4-state HMM for (a) data in nonsensitive region (b) data in intermediate sensitive region (c) data in minor HS (d) data in major HS.

blot. For each validated region, we take the state that covers the most probes as the prediction made by our model. Out of the total 122 site provided to us, which we take as true positive instances, 47 of them are correctly predicted by our algorithm as major hypersensitive sites. Another 32 is found in the minor hypersensitive region predicted by our model. These two sets of sites together account for 65% of the total validated site. We also have another separate set of negative regions consisting of 26 nonsensitive sites. Our model predicted no major or minor HS sites within those regions.

We mentioned in Section 1.1 that one concern that biologists might have is that the underlying biological structure is not changing abruptly. By looking at the raw data in Figure 1.1, for example, we can see that the data look nothing like a piecewise constant function. Rather, we typically observe a gradual change in intensities that form peaks around the hypersensitive regions. Thus one might question the validity of our change point model. Our counter argument is that while it is true that we assumed a model in which the underlying process jumps to another state at each change point, the precise position where this happens can well be characterized by the uncertainty. In Figure 1.8, we plotted the marginal distribution of the change point $p(c_k = i \text{ for some } i|y)$, obtained from 500 sampled segmentations. The uncertainty of the position of the change point is seen from the probability near the inferred change point. A more spreaded "peak" around the change point will indicate a slower change of the underlying process. In our data, we usually have sharp "peaks" implying small uncertainty of the change point position. Some uncertainty in our data is caused by the fact that the neighboring probes are overlapping with each other.

In summary, the results of this preliminary study include three main findings: 1) Hierarchical change point models applied to genome scale studies of DNase I sensitivity/hypersensitivity offer a promising approach to better fit within state variation. This improved fit will be of most value in controlling false positives. 2) These change point models also offer a means to go beyond the geometric models of fragment

length that are implicit in HMMs. In so doing, they provide a principled means to span repeat gaps in tiled array data. 3) These ensemble based estimates provide an assessment of the uncertainty in each inferred state prediction.

1.4 Variants and Extensions of the Basic Model

In previous sections, we presented in detail our new approach for modeling sequence data with continuous observations. In this section, we will present some variants of our model. In particular, we show that we can get a simpler model without length distributions and transition probabilities. If we have prior information on the upper or lower bound of the segment length, this can also be incorporated into our model. Incorporation of this prior information can be important in some applications. Also, in the simpler model, we have much fewer hyperparameters to estimate, this will be useful when we don't have the luxury of extremely long sequences. In section 1.4.2, we address one problem raised in section 1.2.9, where we mentioned that our model selection criterion is conditioned on the parameters estimated. We will discuss how to compute the model probability in a fully Bayesian framework. The method we propose is a sequential updating procedure that is very similar to Sequential Important Sampling (SIS). Actually SIS can be applied to this problem, and we will show how in section 1.4.2. In Section 1.2, the observations were assumed to be i.i.d. given the hidden parameters. In order to show the readers that our model can be much more flexible than the basic model, we present an extension of our model to incorporate Markov dependency within each segment. Higher order Markov can also be accommodated, although we only consider first order Markov here as an illustration. Finally, our model can be easily extended to multiple arrays for our biological application.

1.4.1 A simpler model

We can assume a simpler prior for the segmentation without using explicit length distributions or transition probabilities. This strategy was used in [54], where they assumed a uniform distribution on $1 \dots k_{max}$ for the number of change points, and any segmentations with exactly k change points were assumed to be equally likely. Although they do not have explicit states in their model, we can simply assume that all different assignments of states to segments are equally likely. It is easy to see that a particular segmentation (together with state assignments) with k change points in a sequence of length n has probability $\frac{1}{k_{max}} \frac{1}{\binom{n-1}{k-1}} \frac{1}{D^k}$, where D is the number of states. In this simpler model, we do not need recursion corresponding to (1.4), the main forward recursion is now:

$$
\begin{aligned}
& Pr(y_{1:i} | 1:i \text{ has } k \text{ change point}(k \text{ segments}), \text{ last state is } \delta) \\
= & \sum_{j<i,\delta'} Pr(y_{1:i}, \text{ last change point before } i \text{ is at } j, \text{ with state } \delta' | k, \delta) \\
= & \sum_{j<i,\delta'} Pr(j, \delta', k-1 | i, \delta, k) P(y_{1:i} | k, j, \delta, \delta') \\
= & \sum_{j<i,\delta'} P(j, \delta', k-1 | i, \delta, k) P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \\
= & \sum_{j<i,\delta'} \frac{\binom{j-1}{k-2} D^{k-2}}{\binom{i-1}{k-1} D^{k-1}} P(y_{0:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1)
\end{aligned}
$$

This simpler model can be extended somewhat by taking into account the length constraint and transition constraint. The constraint on the segment length can be represented as an interval $[l_\delta, u_\delta]$, where $l_\delta \geq 1$ is the lower bound and $u_\delta \leq \infty$ is the upper bound for the length of a segment with state δ. We could possibly also have prior constraint on the transition. We can think of this constraint as represented by a graph with each node representing one state. A directed edge is added from state/node δ_1 to δ_2 if the transition from δ_1 to δ_2 is allowed. So the sets of next

possible state when the current state is δ is $N_\delta = \{\delta' : (\delta, \delta') \in E\}$, where E is the set of (directed) edges. In our experience with both HMM and our hierarchical model, the exact value of transition probability is usually much less important in practice than the correct specification of the possible transitions.

With length and transition constraints, the simple combinatorial counting can no longer be used. A recursion is required to count the number of possible segmentations. We let $S(i, \delta, k)$ be the number of possible segmentations (with states) of a sequence of length i, with k change points (k segments), with last state δ. It can be computed using the following recursion:

$$\begin{aligned} S(i, \delta, 1) &= \begin{cases} 1 & \text{if } i \in [l_\delta, u_\delta] \\ 0 & \text{o.w.} \end{cases} \\ S(i, \delta, k) &= \sum_{j<i} \sum_{\delta' \in N_\delta} S(j, \delta', k-1) S(i-j, \delta, 1) \end{aligned}$$

The main recursion now becomes:

$$\begin{aligned} & Pr(y_{1:i} | 1:i \text{ has } k \text{ change points}(k \text{ segments}), \text{ last state is } \delta) \\ = & \sum_{j<i, \delta'} P(j, \delta', k-1 | i, \delta, k) P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \\ & P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \\ = & \sum_{j<i, \delta' \in N_\delta} \frac{S(j, \delta', k-1) S(i-j, \delta, 1)}{S(i, \delta, k)} P(y_{1:j} | \delta', k-1) P(y_{j+1:i} | \delta, 1) \end{aligned}$$

Note that we can define $\hat{P}(y_{1:i} | \delta, k) = S(i, \delta, k) P(y_{1:i} | \delta, k)$ (this is not a probability), so the recursion can be done in terms of $\hat{P}$, which becomes simply

$$\hat{P}(y_{1:i} | \delta, k) = \sum_{j, \delta'} \hat{P}(y_{1:j} | \delta', k-1) \hat{P}(y_{j+1:i} | \delta, 1)$$

Actually, the recursion in Section 1.2 can also be transformed into a simpler form by defining $\hat{P}$ appropriately. With this definition of $\hat{P}$, other formula also become simpler. For example, the recursive backward sampling becomes

$$P(c_{k-1} = j, \delta' | y_{0:n}, c_k = i, \delta) \quad \propto \quad \hat{P}(y_{1:j} | \delta', k-1) \cdot \hat{P}(y_{j+1:i} | \delta, 1)$$

The model we have presented in this subsection has a simpler structure than the basic model we presented in Section 1.2. For example, if we have four states, the model in Section 1.2 has about 40 hyperparameters, while the simpler one here only has 16 parameters. With fewer parameters to estimate, the inference procedure generally becomes more efficient. In particular, if we put a (hyper-hyper-)prior on the hyperparameters, as we will do next, and perform integration over the prior, a 16-dimensional integration is generally much easier to compute numerically than a 40-dimensional one.

1.4.2 Fully Bayesian model selection

In section 1.2.9, we used a small part of data to estimate the hyperparameters, then the expected log likelihood $E \log p(y_{1:n} | \hat{\Theta})$ is approximated using additional data that are not involved in the estimation step, to get an unbiased estimate of the expectation. However, this inference is conditioned on the estimated parameters in the first step, the uncertainty of which is not taken into account. In the fully Bayesian approach, the selection of model is based on the posterior $p(M|y)$, and we choose the model with the largest posterior probability:

$$M^* = \arg\max_M P(M|y)$$

. In Bayesian model selection, we need to specify both the prior $p(M)$ for the model and the priors $p(\Theta|M)$ associated with the parameters for each model. Then the

posterior can be obtained from the Bayes rule:

$$P(M|y) = \frac{p(y|M)p(M)}{p(y)}$$

For prediction problems, a Bayesian would ideally take average across all models to get the marginal probability of new independent observations:

$$p(y^{new}|y) = \sum_{M} p(y^{new}|M)p(M|y)$$

. This is called model averaging, an important technique in Bayesian statistics. Model averaging is usually to expensive to compute, and one often will be satisfied with selecting a single model. Whatever strategy is used, the essential calculation involved in calculating the posterior model probability is to perform the integral $p(y|M) = \int p(y|\Theta, M)p(\Theta|M)d\Theta$. If conjugate prior can be used, the integral can be evaluated in close form. But in problems like in our hierarchical model, we already explored using conjugacy in the specification of the prior for the hidden parameters. The likelihood given the hyperparameters $p(y|\Theta)$ was computed by recursion, not in closed form. Thus it seems impossible to specify appropriate conjugate prior for the hyperparameters. When specification of prior is difficult and when we do not have too much prior information on the hyperparameters, we will usually choose some diffuse prior, trying not to put too much constraint on the hyperparameters, and use numerical methods to evaluated the integral. There are several approaches for this problem.

- We can choose a large enough region and then discretize the region using a regular grid, and use averages over the grid to approximate the integral.

- Modern Bayesian statisticians frequently use MCMC to calculate high-dimensional integrals. Using appropriate proposal distributions for exploring the parameter space, it will generate a Markov chain which converges to the target posterior

distribution.

- Importance sampling is another important tool which also requires an appropriate proposal distribution.

The first approach of computing on the grid does not work for our problem where the dimension of the parameter space is not small. The last two approaches also have problems. In order to find a good proposal distribution, we generally need to have some ideas on the location and dispersion of the posterior distribution. For large data sets, the posterior distribution will usually be concentrated in a small region in the parameter space. Appropriate design of the proposal distribution is critical for the success of the algorithm. For MCMC, this is usually done with trial and error by observing the acceptance rate of the Markov chain. The appropriate proposal distribution for importance sampling seems to be even more difficult to find. The simplest proposal distribution would be the prior $p(\Theta)$ itself. Due to the large discrepancy between the diffuse prior distribution and the posterior, a great deal of computations are wasted on the large regions of the parameter space in the tail of the posterior. In order for importance sampling to work, we must find a proposal distribution that is closer to the posterior. A simple idea is that we start by using the prior as the proposal distribution. As we see more data, we update our knowledge about the posterior and use this as the new proposal for subsequent computations. We will explain this in detail next. As the algorithm progresses, only Θ's that come from the new distribution will be used in the evaluation of subsequent integrals. Those Θ's that come from the prior are only involved in the computation of the first integral below, this increases the efficiency of our algorithm.

Let's first set up the statistical framework for our algorithm. Suppose we have a candidate model M, and a prior on the parameters given M, $p(\theta|M)$, the probability we are interested in is $p(y|M) = \int p(\mathbf{y}|\theta, M)p(\theta|M)\, d\theta$, where $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ are the observations. We assume that $\{y_i\}$ are $i.i.d.$ from the model $p(y|\theta)$. We will assume there is a hidden variable x_i for each y_i, so $p(y_i|\theta)$ is actually the marginal

observations that come from the complete model:

$$p(y_i|\theta) = \sum_{x_i} p(y_i|x_i,\theta)p(x_i|\theta)$$

A remark is in order on what is y_i and x_i for our problem in the previous sections. For our segmentation problem for biological sequence data, the observations are not iid since we sum and integrate over hidden variables. The observations are i.i.d within one segment given the hidden parameters, the mean and variance for that segment. In the application in Section 1.3, we cut the long sequence with hundreds of thousands observations into many sequences each with length about a couple of thousands. Given the hyperparameters, the recursion is done separately for each shorter sequence, and dependence of neighboring sequences were not taken into account. So in order to apply the following procedure, we should think of y_i as containing all observations in one short sequence, and think of x_i as the corresponding segmentation with the hidden parameters for each segment. θ here will correspond to the hyperparameters that we used in the hierarchical model. In this way, we have the i.i.d. data y_i. We set out to compute the marginal distribution of y given a fixed model M. We omit writing M explicitly in the following, with the understanding that each probability is conditioned on a fixed model. One approach of calculating the integral is by straightforward importance sampling:

$$\int p(\mathbf{y}|\theta)p(\theta)\,d\theta \approx \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{y}|\theta^{(i)}), \theta^{(i)} \sim p(\theta)$$

If n is large, the overlap between the prior $p(\theta)$ and the likelihood $p(\mathbf{y}|\theta)$ is typically small, the above computation will be inefficient, because most of the computation will be wasted on the region of θ for which $p(y|\theta)$ is extremely small and contribute little to the integration. We propose the following strategy:

1. partition the data into K parts: $\cup_{k=1}^{K}\mathbf{y}_k = \{y_1, \ldots, y_n\}$

2. Set the initial proposal distribution to be the prior: $p_0(\theta) := p(\theta)$, draw N samples from $p_0(\theta)$, $\{\theta_0^{(i)}\}_{i=1}^N$

3. for t from 1 to K

   - compute $m_t := \int p(\mathbf{y}_t|\theta)p_{t-1}(\theta)\,d\theta$, using samples from $p_{t-1}(\theta)$ as an approximation: $\frac{1}{N}\sum_{t=1}^N p(\mathbf{y}|\theta_{t-1}^{(i)})$. Note m_i is actually the probability $p(\mathbf{y_t}|\mathbf{y_1},\ldots,\mathbf{y_{t-1}})$
   - Define $p_t(\theta)$ to be the posterior distribution of θ give $\{\mathbf{y_1},\ldots,\mathbf{y_t}\} : p_t(\theta) \propto p(\mathbf{y}_t|\theta)p_{t-1}(\theta)$. Use samples $\{\theta_t^{(i)}\}$ to represent $p_t(\theta)$, which can be done in 3 steps:
     - since $p_{t-1}(\theta)$ is represented by samples $\{\theta_{t-1}^{(i)}\}$, $p_t(\theta)$ can be represented as weighted samples $(\theta_{t-1}^{(i)}, w_t^{(i)})$, where $w_t^{(i)} = p(\mathbf{y_t}|\theta_{t-1}^{(i)})$
     - resample from $\{(\theta_{t-1}^{(i)}, w_t^{(i)})\}$ to obtain $\{\tilde{\theta}_t^{(i)}\}$
     - perform one Gibbs sampling-like step: for each i, draw $\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_t}\}$ from the conditional distribution $p(\mathbf{x_1},\ldots,\mathbf{x_t}|\tilde{\theta}_t^{(i)},\mathbf{y_1},\ldots,\mathbf{y_t})$. Then conditioned on the $\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_t}\}$ just obtained, draw $\theta_t^{(i)}$ from the conditional distribution $p(\theta_t^{(i)}|\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_t}\})$

   endfor

4. output $\prod_{i=1}^K m_i$ as the approximation to the integral $\int p(\mathbf{y}|\theta)p(\theta)\,d\theta$

The correctness of the algorithm can be seen simply by that $m_t \approx p(\mathbf{y_t}|\mathbf{y_1},\ldots,\mathbf{y_{t-1}})$

In the procedure above, each step of the integral is still performed with importance sampling. The advantage of this procedure is that after each data set comes in, we use that to update the posterior distribution, which is then used as the proposal distribution in the next step to compute m_t. This importance sampling works because

in the straightforward approach, we use samples from the prior $p(\theta)$ to compute the integral $\int p(\mathbf{y}|\theta)p(\theta)d\theta$, here we use samples from $p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})$ to compute the integral $\int p(\mathbf{y_t}|\theta)p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})d\theta$. The difference between $p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})$ and $p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_t})$, is supposedly much less than the difference between the prior $p(\theta)$ and the posterior $p(\theta|\mathbf{y})$ conditioned on all data. This is what will make our algorithm more efficient.

Actually, we do not have to compute the m_t's on the way. At the end of the algorithm, what we obtained are samples $\{\theta_K^{(i)}\}$ from the posterior distribution $p(\theta|\mathbf{y})$. In the next subsection, I will show how to evaluate the integral $\int p(\mathbf{y}|\theta)p(\theta)d\theta$ if we can obtain samples from the posterior $p(\theta|\mathbf{y})$, without the requirement to be able to evaluate $p(\theta|\mathbf{y})$ (which can be obtained only if the integral $\int p(\mathbf{y}|\theta)p(\theta)d\theta$ can be evaluated first).

Consider our hierarchical model, in step 3 of the above procedure, sampling from $p(\theta_t^{(i)}|\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\})$ is generally easy since x_t contains information about the segmentation, together with the hidden parameters. The difficulty resides in sampling from $p(\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}|\theta)$, which requires us to do forward recursion again with new θ on the data $\{\mathbf{y_1}, \ldots, \mathbf{y_t}\}$, then do backward sampling to get samples for $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$. To avoid this, we can perform only a half step of step 3, so samples $\mathbf{x_t}$ is drawn only at the tth time step. Now our new algorithm will looks like the following:

1. partition the data into K parts: $\cup_{k=1}^{K} \mathbf{y}_k = \{y_1, \ldots, y_n\}$

2. Set the initial proposal distribution to be the prior: $p_0(\theta) := p(\theta)$, drawing N samples from $p_0(\theta)$, $\{\theta_0^{(i)}\}_{i=1}^{N}$

3. for t from 1 to K

   - Obtain samples $\{\theta_t^{(i)}, \mathbf{x_1^{(i)}}, \ldots, \mathbf{x_t^{(i)}}\}$ representing the distribution $p(\theta, \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}|\{\mathbf{y_1}, \ldots, \mathbf{y_t}\})$. This can be done in 3 steps:

- for each i from 1 to N,draw samples $\mathbf{x_t^{(i)}}$ from $p(\mathbf{x_t}|\theta_{t-1}^{(i)}, \mathbf{y_t})$, so $p(\theta, \mathbf{x_1}, \ldots, \mathbf{x_t}|\mathbf{y_1}, \ldots, \mathbf{y_t})$ is represented by $\{\theta_{t-1}^{(i)}, \mathbf{x_1^{(i)}}, \ldots, \mathbf{x_t^{(i)}}, w_t^{(i)}\}$ where $w_t^{(i)} = p(\mathbf{y_t}|\theta_{t-1}^{(i)})$
- resample from $\{\theta_{t-1}^{(i)}, \mathbf{x_1^{(i)}}, \ldots, \mathbf{x_t^{(i)}}, w_t^{(i)}\}$ to obtain $\{(\tilde{\theta}_t^{(i)}, \mathbf{x_1^{(i)}}, \ldots, \mathbf{x_t^{(i)}})\}$
- Conditioned on the $\{\mathbf{x_1^{(i)}}, \mathbf{x_2^{(i)}}, \ldots, \mathbf{x_t^{(i)}}\}$ obtained, draw $\theta_t^{(i)}$ from the conditional distribution $p(\theta_t^{(i)}|\{\mathbf{x_1^{(i)}}, \mathbf{x_2^{(i)}}, \ldots, \mathbf{x_t^{(i)}}\})$

endfor

4. use the samples $\{\theta_K^{(i)}\}$ to compute the integral $\int p(\mathbf{y}|\theta)p(\theta)\, d\theta$, using the method to be presented in section 1.4.3.

In the above, it might not be obvious to the readers why we have the weights $w_t^{(i)} = p(\mathbf{y_t}|\theta_{t-1}^{(i)})$. Actually, the above procedure is almost identical to the sequential importance sampling procedure with resampling, the only (important) difference is that we periodically redraw θ's at each time step. If we skip this step, the resampling step will make the number of non-unique θ's smaller and smaller, which will be a problem when we later use these $\theta' s$ to compute the integral. After realizing the connection with SIS, the weights can be computed from the general formula in [53]. From equation (2) in section 2.1 of their paper, we get that the appropriate weight is

$$w_t^{(i)} = \frac{p(\theta_{t-1}^{(i)}, \mathbf{x_1}, \ldots, \mathbf{x_t}|\mathbf{y_1}, \ldots, \mathbf{y_t})}{p(\theta_{t-1}^{(i)}, \mathbf{x_1}, \ldots, \mathbf{x_{t-1}}|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}}|)p(\mathbf{x_t}|\theta_{t-1}^{(i)}, \mathbf{y_t})} \propto p(\mathbf{y_t}|\theta_{t-1}^{(i)})$$

We have not explained how we should partition the data in step 1. This can be done in many ways. A good strategy might be to partition the data such that the size of the tth partition is a fraction, say 1/2, of the $t+1$th partition. Since we use the posterior $p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})$ to calculate the integral $\int p(\mathbf{y_t}|\theta)p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})d\theta$:

$$\frac{1}{N}\sum_{i=1}^{N} p(\mathbf{y_t}|\theta_{t-1}^{(i)}) \quad \theta_{t-1}^{(i)} \sim p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})$$

if we partition the data this way, the information about θ contained in the previous posterior $p(\theta|\mathbf{y_1}, \ldots, \mathbf{y_{t-1}})$ is roughly the same as the information contained in the likelihood $p(\mathbf{y_t}|\theta)$ in the next step.

1.4.3 Calculating marginal probability from posterior samples

This subsection is more generally applicable than the previous one. Basically it provides a method to calculate the marginal probability in a Bayesian framework when we can obtain posterior samples. So the goal is to compute $\int p(y|\theta)p(\theta)d\theta$ given samples from the posterior $p(\theta|y)$. A simple strategy using importance sampling does not work. If we write the integral as follows:

$$\int p(y|\theta)p(\theta)d\theta = \int \frac{p(y|\theta)p(\theta)}{p(\theta|y)} \cdot p(\theta|y)d\theta$$

and try to evaluate the integral using samples $\{\theta^{(i)}\}_{i=1}^{N}$ from the posterior with

$$\frac{1}{N}\sum_i \frac{p(y|\theta^{(i)})p(\theta^{(i)})}{p(\theta^{(i)}|y)}$$

this does not work because we don't know how to evaluate $p(\theta|y)$, which requires that we are able to compute the normalizing constant, which is exactly $\int p(y|\theta)p(\theta)d\theta$, the integral we want to compute in the first place. A simple strategy will solve this problem.

Generally, if we want to compute some integral $\int f(\theta)p(\theta)d\theta$, the (unnormalized) importance sampling works as follows:

- draw samples $\{\theta^{(i)}\}$ from a proposal distribution $q(\theta)$
- calculate the importance weights for each sample: $w^{(i)} = \frac{p(\theta)}{q(\theta)}$
- use $\frac{\sum w^{(i)} f(\theta^{(i)})}{\sum w^{(i)}}$ as an approximation to the integral.

The main advantage of using this formula instead of the more direct sample average estimate $\frac{1}{N}\sum w^{(i)} f(\theta^{(i)})$ is that the distribution $q(\theta)$ does not have to be normalized. Aided by this observation, we can now write our integral as

$$\int p(y|\theta)p(\theta)d\theta = \int p(y|\theta)\frac{p(\theta)}{p(\theta,y)}p(\theta,y)d\theta$$

where we used the unnormalized distribution $p(\theta, y)$ instead of $p(\theta|y)$. By the formula above, we can calculate the weights $w^{(i)} = \frac{p(\theta^{(i)})}{p(\theta^{(i)},y)} = \frac{1}{p(y|\theta^{(i)})}$, and the integral can then be approximated by

$$\frac{\sum w^{(i)} p(y|\theta^{(i)})}{\sum w^{(i)}} = \frac{\sum \frac{1}{p(y|\theta^{(i)})} p(y|\theta^{(i)})}{\sum \frac{1}{p(y|\theta^{(i)})}} = \frac{N}{\sum \frac{1}{p(y|\theta^{(i)})}}$$

So the integral is approximated by the harmonic mean of $p(y|\theta^{(i)})$, which does not require evaluating the posterior probability.

1.4.4 Markov dependence

Our basic model in Section 1.2 assumes that each segment has its own mean and variance, and each observation within one segment are generated i.i.d. given these hidden parameters. In this subsection, we will present an extension that takes into account higher order Markov dependency within one segment. Here we only consider one particular Markov model that is called autoregressive model which has been studied a lot in parametric time series theory. Time series is a well-developed statistical theory which has many applications in economics and financial modeling. A good introductory book on the classical theory for mathematically inclined readers is [11], while the more recent book [28] focuses more on the nonparametric aspect of the theory. An autoregressive model of order p, $AR(p)$, is defined as

$$y_t = \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \epsilon_t \tag{1.12}$$

where in general ϵ_t is a white noise process $WN(0, \sigma^2)$, i.e.

$$E(\epsilon_t) = 0,\ Var(\epsilon_t) = \sigma^2,\ \text{ and }\ Cov(\epsilon_t, \epsilon_s) = 0,\ \text{ for all } t \neq s$$

We assume the more strict condition that ϵ_t are i.i.d. $N(0, \sigma^2)$. By the above definition, an i.i.d. model we studied before can be considered as the special case with order p equal zero. Next we illustrate the model with $p = 1 : \ y_t = \beta y_{t-1} + \epsilon_t$. $AR(p)$ with $p > 1$ can be studied with a little bit more complicated vector algebra using a multidimensional normal prior similar to the Bayesian linear regression in [34]. We also assume that the time series has mean zero for simplicity. If a stationary $AR(1)$ model has mean μ, then we have that $y_t - \mu = \beta(y_{t-1} - \mu) + \epsilon_t$, i.e. $y_t = \mu(1 - \beta) + \beta y_{t-1} + \epsilon_t$, thus a series with non-zero mean can be modeled by an extra intercept coefficient $\beta_0 := \mu(1 - \beta)$.

For technical reasons, in a $AR(1)$ model, we usually assume that $|\beta| < 1$, which ensures that model (1.12) admits a unique stationary and causal solution, although this constraint is not going to be put on our model, since we will later assume a Gaussian prior for β, which has infinite support. The AR(1) model exhibits dramatically different behavior depending on the sign of β. It can be shown that β is actually the correlation coefficient $Corr(y_t, y_{t-1})$. Thus with negative β, the time series has an oscillatory behavior, while with positive β, the series looks smoother. Fig 1.9 plotted two time series with $\beta = 0.9$ and $\beta = -0.9$ respectively. This illustrated that we can possibly segment data into regions according to this behavior. We are currently not aware of any application of this model, so this subsection mainly serves a pedagogic purpose to illustrate that our model can be much more flexible than presented in Section 1.2.

The model is same as before, except now we assume that for a segment $[i, j]$, given

the hidden parameters β and σ^2, the observations are generated by

$$\begin{aligned} y_i &= \epsilon_i \\ y_t &= \beta y_{t-1} + \epsilon_t, t = i+1, \ldots, j \\ \epsilon_t &\overset{iid}{\sim} N(0, \sigma^2), t = i, \ldots, j \end{aligned}$$

We put the following prior on the hidden parameters, the same as used in the basic model:

$$\beta_i | \sigma_i^2, A \sim N(\beta_{\delta_i}^{(h)}, \frac{\sigma_i^2}{k_{\delta_i}^{(h)}})$$

$$\sigma_i^2 | A \sim Inv\chi^2(\nu_{\delta_i}^{(h)}, {\sigma_{\delta_i}^2}^{(h)})$$

The probability $p(y_{i:j}|\delta, 1)$ can be computed analytically as before (omitting superscript and subscript for the hyperparameters in the following):

$$\begin{aligned} p(y_{i:j}|\delta, 1) &= \int\int \frac{1}{\sqrt{2\pi{\sigma^2}^n}} e^{-\frac{y_i^2 + \sum_{t=i}^{j}(y_{t+1} - \beta y_t)^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi\frac{\sigma^2}{k}}} e^{-\frac{k(\beta - \beta_\delta)^2}{2\sigma^2}} \cdot \\ &\quad \frac{\frac{\nu}{2}^{\nu/2}}{\Gamma(\frac{\nu}{2})} \sigma_\delta^\nu \sigma^{-2(\frac{\nu}{2}+1)} e^{-\frac{\nu\sigma_\delta^2}{2\sigma^2}} d\beta d\sigma^2 \end{aligned}$$

First the integration over β can be computed,

$$\begin{aligned} & \int e^{-\frac{y_i^2 + \sum_{t=i}^{j}(y_{t+1} - \beta y_t)^2}{2\sigma^2}} e^{-\frac{k(\beta - \beta_\delta)^2}{2\sigma^2}} d\beta \\ = & \sqrt{2\pi \frac{\sigma^2}{k_\delta + j - i + 1}} e^{-\frac{\lambda'}{2\sigma^2}} \end{aligned}$$

where λ' depends on the observations and the hyperparameters. So instead of λ in Section 1.2, we have a different λ'. The corresponding formula for $p(y_{i:j}|\delta, 1)$ is exactly as in (1.3), with λ replaced by λ'. It is obvious from the discussion of Section 1.2 that all other recursions remain exactly the same.

From the above discussion, it is evident that our model has great flexibility in

modeling the dependency using an embedded Bayesian time series model. The only change we need to make is in computing the $p(y_{i:j}|\delta, 1)$. Thus if the program is written in separate modules, the programming burden when we change to higher order Markov model is minimal, with only one module need to be changed. No other parts of the code need to be modified at all.

1.4.5 Multiple arrays

In the application presented in the last section, we only used one sequence. The extension to the case where we have multiple replica arrays of the same experiment is straightforward. First we obtain the hyperparameters using EM as before. The distributions on transitions and segment lengths are constrained to be the same across replicas, since they reflect the underlying biological structure that should be kept the same across different arrays. But the other hyperparameters are fitted independently for each replica. The variations between the replicas are reflected in the difference of those hyperparameters. This can be regarded as a normalization method for different arrays. If the arrays have been normalized beforehand, we can use only one set of hyperparameters. Generally, for m replicas, we will have m sets of hyperparameters $\{\Theta^{(r)}\}_{r=1}^{m}$. Assuming independence of observations between the replicas, the probability of observation can be factored as $p(\{y_{i:j}^{(r)}\}_{r=1}^{m}|\delta, 1) = \prod_{r=1}^{m} p(y_{i:j}^{(r)}|\delta, 1, \Theta^{(r)})$. The recursions and sampling is same as before, replacing $p(y_{i:j}|\delta, 1)$ with $p(\{y_{i:j}^{(r)}\}_{1}^{m}|\delta, 1)$.

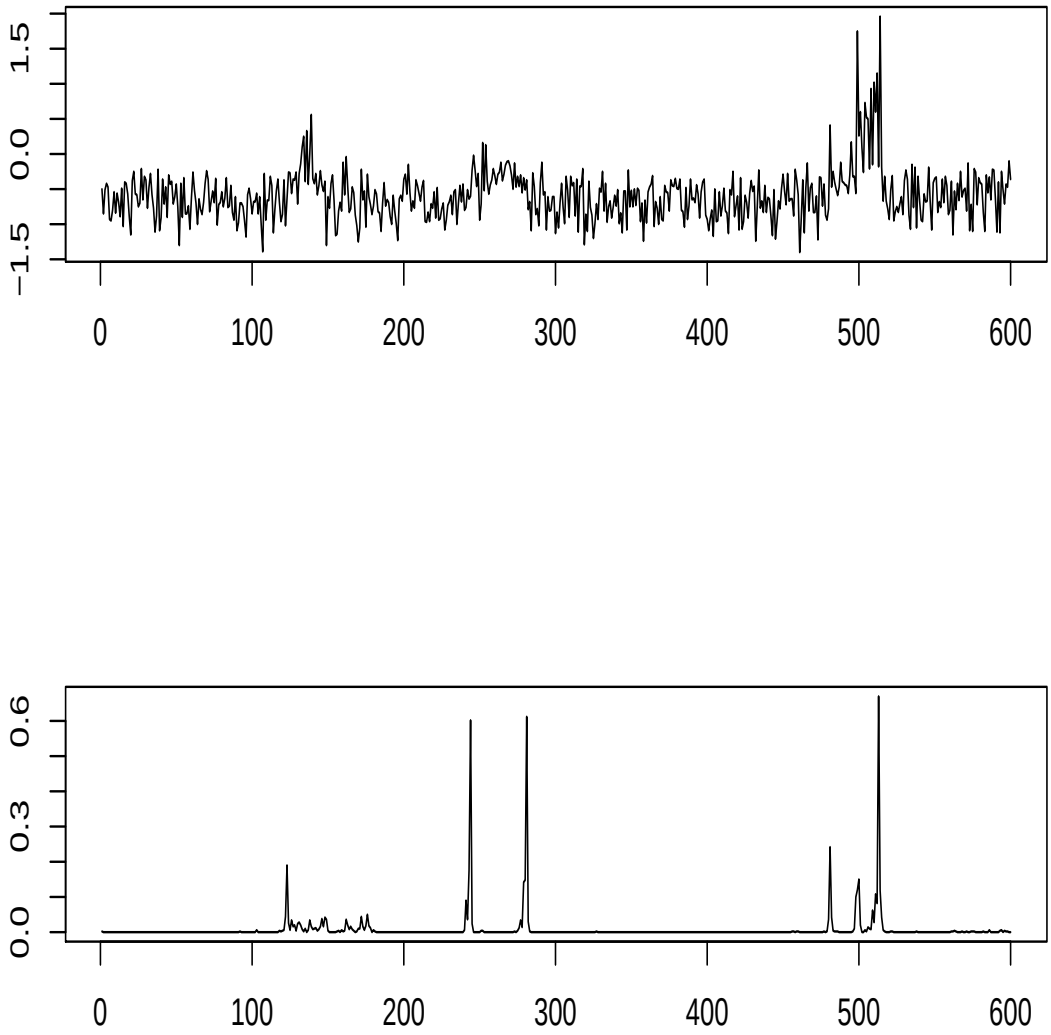


Figure 1.8: Top: sequence data. Bottom: probability of change.

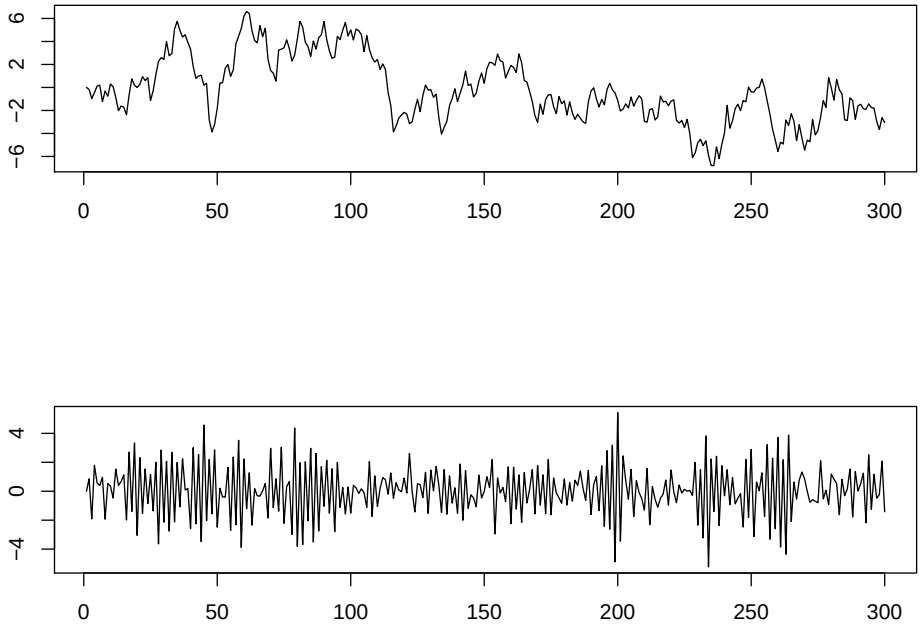


Figure 1.9: Times series simulated from model $AR(1)$ with standard Gaussian noise. Top: $\beta = 0.9$. Bottom: $\beta = -0.9$.

Chapter 2

Variational Image Superresolution

Super-resolution is an important but difficult problem in image/video processing. If a video sequence or some training set other than the given low-resolution image is available, this kind of extra information can greatly aid in the reconstruction of the high-resolution image. The problem is substantially more difficult with only a single low-resolution image on hand. The image reconstruction methods designed primarily for denoising is insufficient for super-resolution problem in the sense that it tends to oversmooth images with essentially no noise. In this chapter, we propose a new adaptive linear interpolation method based on variational method and inspired by local linear embedding (LLE). The experimental result shows that our method avoids the problem of oversmoothing and preserves image structures well.

2.1 Introduction

Super-resolution is an image processing technique to obtain high-resolution image from its low-resolution counterpart, either using information from a single image or image sequence. Most of the literature focus on the latter [43][74]. With only one image, it is generally difficult to interpolate the missing pixels with little prior information about the image.

Many approaches have been proposed in recent years. For video/image sequence, information on subpixel motion between temporally adjacent images can be utilized. Due to the ill-posedness of the super-resolution problem, regularization[43] or a Bayesian framework[94] usually needs to be set up, the two approaches being closely related.

For only one single image, traditionally the most often used method is bilinear or bicubic interpolation, which works under the assumption that images are smooth for the most part. One recent advance on single image super-resolution is the work by Freeman et al[33]. Using a large database of image patches consisting of low-resolution and high-resolution pairs, the algorithm will search the database for the

corresponding high-resolution patch, for each of the image patches in the given low-resolution image. One obvious disadvantage of this approach is that it requires the existence of a database in the first place.

We propose a novel algorithm for single image super-resolution, without the need for a database or other training images. Our algorithm is inspired by Local Linear Embedding (LLE) used for dimension reduction. By trying to express each pixel value as a linear combination of neighboring pixel values, we can estimate the local structure of the low-resolution image which we can then use to construct a filter for interpolation in the second step.

One work similar in spirit to ours is [20]. There the authors also try to express the center pixel as a linear combination of its neighbors. Similar to Freeman[33], their algorithm also requires a training set of pairs of patches in order to estimate those linear coefficients. And the coefficients are learned in the regression framework, giving as a result a spatially adaptive filter. There algorithm can be regarded as a complement to [33] which works well on cartoon images.

One approach to single image super-resolution without training, which is also related to ours, is mentioned in [15] using digital TV filter[14], which is the discrete counterpart of the variational functional method with a total variation (TV) norm. The digital TV filter is originally designed with the goal of image denoising in mind. Although the authors demonstrated that using a TV smoothing term can largely preserve edges and other important features, the result still looks blurred and blocky (using higher order smoothness term can alleviate the second problem with the cost of exacerbating the first problem). We think that directly using denoising algorithm on the super-resolution problem is somewhat unnatural, with no special attention to the problem at hand. Our algorithm uses variational approach on the linear coefficients instead, so the smoothing term has an effect on the local filter, not directly on the image.

2.2 local structure estimation

Denoting the given image by u, defined on $\Omega = [0,1] \times [0,1]$. We start by assuming that each point $x \in \Omega$ can be approximately expressed as an weighted combination of its neighbors (eight neighbors are used in our implementation):

$$u(x) \approx \sum_{i=1}^{8} w^{(i)}(x) u(x + \delta_i) \tag{2.1}$$

, where $x+\delta_i$ are the neighbors of x, with $\delta_1 = (-h, -h)$ etc. For grayscale images, for example, the above equality is trivially satisfied for an infinite number of combinations of $w^{(i)}, i = 1..8$, given the image u. So trying to estimate w separately for each point x is futile without further constraint. Here we use the variational approach with a TV norm proposed in [72]:

$$\min_{w} \int \left(\sum_{i} w^{(i)} u(\cdot + \delta_i) - u \right)^2 dx + \lambda \sum_{i=1}^{8} \int |\nabla w^{(i)}| dx \tag{2.2}$$

,where λ is a parameter that must be set to balance the fidelity term and the smoothness term. This minimization problem can be solved iteratively using the corresponding evolution PDE:

$$w_t^{(i)} = div \left(\frac{\nabla w^{(i)}}{|\nabla w^{(i)}|} \right) - \lambda \left(\sum_{i} w^{(i)} u(\cdot + \delta_i) - u \right) u(\cdot + \delta_i) \tag{2.3}$$

,which can then be discretized and solved numerically.

2.2.1 Connections to LLE

Given a set of N vectors $\{X_i\}$ in high dimensions. The LLE solves the following minimization problem:

$$\min_{W} \sum_{i} (X_i - \sum_{j} W_{ij} X_j)^2 \tag{2.4}$$

with the constraint

$$\sum_j W_{ij} = 1, W_{ij} \geq 0 \tag{2.5}$$

where the index j runs over those of X_j that is among the k nearest neighbors of X_i. with nearest neighbors determined by some metric $d(X_i, X_j)$. This minimization problem has a non-trivial solution since it is usually assumed that the dimension of X_i is much bigger than k.

To generalize LLE, we first assume that the data come in with two components $X_i = (Y_i, Z_i)$ (think of Y_i as grid position, and Z_i as intensity value). Now we can minimize the following:

$$\min_W \sum_i (Z_i - \sum_j W_{ij} Z_j)^2 \tag{2.6}$$

the index j still runs over k nearest neighbors of X_i. but now with nearest neighbors determined by some metric $d(Y_i, Y_j)$ depending on the other component of X.

If dimension of X_i is small compared to k (as in the case of an image), we must add regularization term to make the problem well-posed. And we will recover the discrete counterpart of (2.2) after ignoring the convexity constraint(2.5).

2.2.2 Comparison with Digital TV Filter

The digital TV filter also implicitly estimates some kind of local linear structure. It repeatedly applies the following filtering step on the current image until convergence (see equation (9) in [14]):

$$u^{t+1}(x) = \sum_{i=1}^{8} h^{(i)}(x) u^t(x + \delta_i) + h^{(0)}(x) u^0(x) \tag{2.7}$$

, where u^t is the reconstructed image at the t-th iteration, and the coefficients h are based on local image gradient and must be recomputed for each iteration. Note one difference in form between (2.1) and (2.7) is that we don't have a corresponding $w^{(0)}$ in (2.1). The reason is that if $w^{(0)}$ were added, (2.2) would have a trivial solution, with

$w^{(0)} \equiv 1$, $w^{(i)} \equiv 0, i = 1 \ldots 8$. The coefficients for digital TV filter after convergence is quite different from the coefficients estimated from our method (Fig. 2.1). This is due to the fact that we estimate the coefficients using a global functional, and the TV filter computes the coefficients using local information only. Another reason for the difference is that our approach estimates orientation-specific weight, the digital TV only estimates local gradient magnitude.

2.3 super-resolution

Suppose now we are given an $n \times n$ low-resolution image u^{low}, and have also obtained an initial reconstructed $zn \times zn$ high-resolution image $\hat{u}_0$ (by nearest neighbor interpolation, for example), where z is the magnification ratio. Using variational method outlined in the last section on u^{low}, we get eight $n \times n$ weight matrices $w^{(i)}$ representing roughly the local structure in the low-resolution image. These weight matrices cannot be directly used as filters for high-resolution image because it is smaller in size. To obtain suitable $zn \times zn$ weight matrices $\tilde{w}^{(i)}$, we can use linear interpolation on the low-resolution weight matrices to obtain the high-resolution weight matrices. Finally, the weight matrices can act as adaptive filters for interpolation:

$$u^{high}(x) = \sum_{i=1}^{8} \tilde{w}^{(i)}(x)\hat{u}_0(x + \delta_i) \tag{2.8}$$

2.4 experimental result

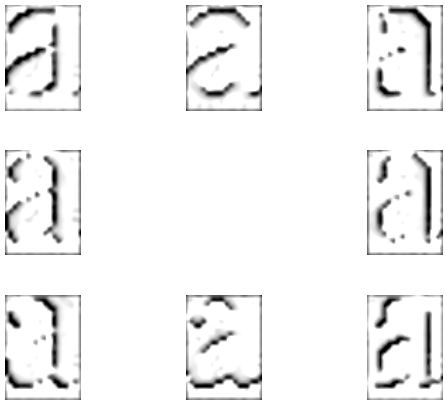
We compare the proposed method with both bicubic interpolation and digital TV filter using grayscale images. We use $z = 3$ in the following experiments. The smoothing parameter used is $\lambda = 0.1$ in all our experiments. It is observed that the result is quite similar for all different smoothing parameters we tried. The reason is probably that the smoothness parameter affect the smoothness of the weights, and only has indirect effect on image itself through the weights. We first apply the algorithm to

the house image (Fig. 2.2). Both bicubic interpolation and digital TV filter oversmooth the image. To make this point clearer, the Sobel edge detector is applied on the results using the same threshold. The edge detector returns less edges on both bicubic interpolated and TV filtered images, which shows the oversmoothing problem of both methods. Another advantage of our method over bicubic interpolation is that the edges are less wiggled, as can be seen from the roof of the house image in (c) of Fig 2.2, for example.

As a second example, the fingerprint image is used to see how well our method works for texture images. It is well known that TV filter does not work on texture images as can be seen from Fig 2.3. Our method also gives reasonable result in this case.

2.5 Conclusion

In this work, we proposed a new algorithm for single-image super-resolution problem using variational method. Instead of working on the image space as in the previous work utilizing variational method, we use variational formulation to estimate the local structure of an image. The resulting adaptive filter reflects both local pixel variance and global image information. The experimental result shows some advantage of our method over some previous approaches. A future research direction might be to explore other applications of the variational estimation of the local image structure.

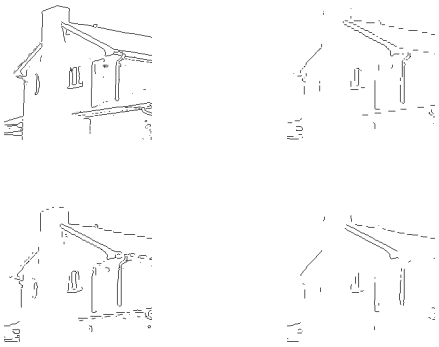


(a) $w^{(i)}$ computed in our method

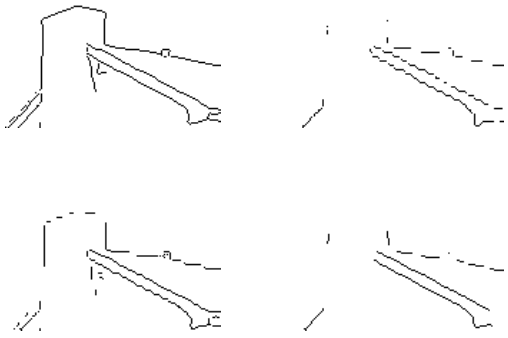
(b) $h^{(i)}$ for TV filter after convergence

Figure 2.1: The linear coefficients computed in our method and digital TV filter is quite different for a simple image

(a) interpolation results

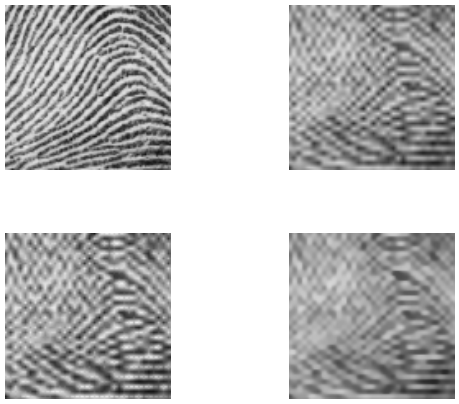


(b) response to Sobel edge detector

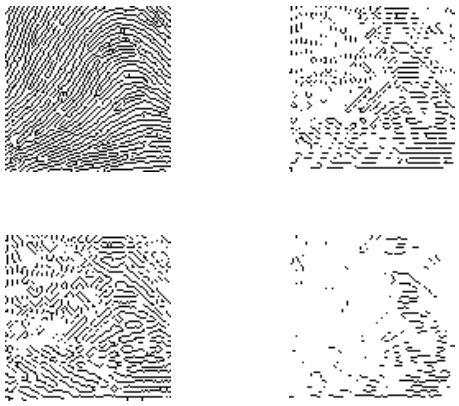


(c) zoom-in of part in (b)

Figure 2.2: Comparison of interpolation results. Upper Left: original high-resolution image. Upper Right: bicubic interpolation, PSNR=26.9. Lower Left: our method, PSNR=27.6. Lower Right: digital TV filter, PSNR=26.9.



(a) interpolation results



(b) response to Sobel edge detector

Figure 2.3: Comparison of results on fingerprint image. Upper Left: original high-resolution image. Upper Right: bicubic interpolation, PSNR=16.3. Lower Left: our method, PSNR=16.8. Lower Right: digital TV filter, PSNR=16.2.

Chapter 3

Nonparametric Bayesian Asymptotics

Bayesian inference distinguishes itself from the frequentist school by its explicit quantification of uncertainty of the parameter with prior specification. The classical approach uses subjective priors elicited from field experts and domain knowledge. The modern Bayesian school have instead shifted attention more towards the construction of priors using formal rules in the hope of dealing with arbitrariness of the prior.

Asymptotics for infinite-dimensional Bayesian statistics has been receiving a lot of attention recently. In these studies, the Bayesian inference is approached from a frequentist point of view, that is, we assume there is a true underlying probability distribution that generates the data. Naturally one desired property is that as more and more observations are made from the underlying generating mechanism, we will obtain accurate estimate of the true distribution. While traditional Bayesians do not believe in such an assumption, it is shown by [9] that this property is the same as intersubjective agreement, which means two Bayesians will eventually come to roughly the same conclusion after seeing enough data.

The posterior distribution typically behaves well under regular parametric models. Doob showed that consistency is achieved under almost no assumptions on the model, except for a zero measure set under the prior, although in topological terms this set can be large. For infinite-dimensional models, however, the matter is more subtle. Strange behavior can be observed under some priors as documented in [22]. Given the prior Π on the set $\mathcal{P}$ of probability distribution, the posterior is a random measure:

$$\Pi^n(B|X_1,\ldots,X_n) = \frac{\int_B \prod_{i=1}^n p(X_i) d\Pi(P)}{\int \prod_{i=1}^n p(X_i) d\Pi(P)}$$

For ease of notation, we will omit the conditioning and only write $\Pi^n(B)$ for the posterior distribution. We say that the posterior is consistent if

$$\Pi^n(P \in \mathcal{P} : d(P, P_0) > \epsilon) \to 0 \text{ in } P_0^n \text{ probability.}$$

where P_0 is the true distribution and d is some suitable distance function between probability measures.

To study rates of convergence, let ϵ_n be a sequence decreasing to zero, we say the rate is at least ϵ_n if for sufficiently large constant M

$$\Pi^n(P : d(P, P_0) \geq M\epsilon_n) \to 0 \text{ in } P_0^n \text{ probability.}$$

We can also have a slightly weaker definition of rates of convergence by replacing M with a sequence M_n and requiring that the above posterior mass converge to zero for any sequence M_n that diverges to infinity.

On the positive side, [75] shows consistency for specific distributions by constructing a sequence of tests of the true distribution against distributions some positive distance away. The tests can trivially be constructed for weak neighborhoods. The construction of similar tests for stronger topology (typically measured in Hellinger distance, for example) is not so straightforward and requires extra works. [6] gives sufficient conditions that guarantee consistency of infinite-dimensional models by bounding the likelihood ratio under bracketing entropy constraint on sieves. [79] studied rates of convergence. A related approach by constructing a sequence of tests appeared in [38].

The conditions imposed in the above are sufficient but not necessary. It is important to see to what extent these conditions can be relaxed. Another line of work parallel to the development above by Stephen Walker and his collaborators proves consistency and rates of convergence under slightly less stringent conditions. These results are established by constructing a certain supermartingale and consistency and rates of convergence is shown by focusing on the distance of certain predictive distributions to the true one. This approach does not require construction of sequence of tests or sieves. It is shown that this new approach can lead to somewhat weaker sufficient conditions or faster rates.

In the first section of this chapter, we extend the existing theory to the non i.i.d. case where we consider the consistency problem of function approximation using step functions. A concrete prior is constructed to illustrate our assumptions. In the second section of this chapter, we provide a simple proof of the rates of convergence of posterior distribution under model misspecification, which was previously studied in [49].

3.1 Consistency of Bayesian Approximation with Step Functions

3.1.1 Introduction

Step functions are frequently studied in the statistical literature, either as an approximation to other functions or as a true model in itself. For example, [42] used a step function as underlying intensity in a Poisson process. Numerous papers on change point analysis also bear similarity with step functions, although the emphasis is usually on the location of the change point.

A study of some nonparametric Bayesian regression problems appears in [92], where they use random covariates assumed to be i.i.d. We will prove a similar result for the special simple case that the underlying function are piecewise constant with a finite number of jumps, corrupted by i.i.d. Gaussian noise with known variance. A main difference from the previous problems studied is that we assume that the covariates are on a regular grid in the interval $[0, 1]$, so the observations are independent but not identically distributed. We show that if the prior is such that it sufficiently penalizes frequent jumps, the Bayesian procedure is consistent in the sense that the posterior probability mass will concentrate on a L_2 neighborhood of the true function. We will construct a simple prior which shows that the assumptions we made for consistency are easily satisfied in practice.

In an influential paper, [6] showed how to prove consistency in nonparametric Bayesian density estimation. The main idea is to express the posterior as a ratio, then show that the numerator is exponentially small, while the denominator cannot decrease exponentially fast. We will follow this reasoning in the current work.

3.1.2 Main Results

We closely follow the proofs in [6]. For that reason, we only show the part of proofs that is different from [6]. Also, we will make use of the notations in [6] as much as possible so the readers can easily find corresponding results in that paper.

The data we observe are $y_i = f_0(x_i) + \epsilon_i, x_i = \frac{i}{n}, \epsilon_i \overset{iid}{\sim} N(0, \sigma^2), i = 0, 1, \ldots, n$, where σ^2 is assumed to be known. The true underlying step function f_0 is assumed to come from the set

$$\begin{aligned}\mathfrak{F} \;=\; & \{f : f \text{ is a step function defined on } [0,1] \text{ with finite number of jumps,} \\ & \text{and } ||f||_\infty := \sup |f(x)| < K\}\end{aligned}$$

The posterior distribution is given by

$$\Pi^n(A) = \frac{\int_A \prod_{i=0}^n e^{-\frac{(y_i - f(x_i))^2}{2\sigma^2}} \Pi(df)}{\int \prod_{i=0}^n e^{-\frac{(y_i - f(x_i))^2}{2\sigma^2}} \Pi(df)} = \frac{\int_A \prod_{i=0}^n e^{-\frac{(y_i - f(x_i))^2 - (y_i - f_0(x_i))^2}{2\sigma^2}} \Pi(df)}{\int \prod_{i=0}^n e^{-\frac{(y_i - f(x_i))^2 - (y_i - f_0(x_i))^2}{2\sigma^2}} \Pi(df)} =: \frac{m_n(y^n, A)}{m_n(y^n)}$$

We construct a sequence of subsets of $\mathfrak{F}$

$$\mathfrak{F}_n = \{f \in \mathfrak{F} : f \text{ has at most } k_n \text{ jumps}\}$$

and consider a small neighborhood of f_0 defined by

$$A_\epsilon = \{f \in \mathfrak{F} : \int_0^1 (f - f_0)^2 \, dx \leq \epsilon^2\}$$

Denoting the prior on $\mathfrak{F}$ by $\Pi(df)$, the assumptions that we use in the following proofs

are

Assumption 3.1 *For every* $\epsilon > 0, \Pi(A_\epsilon) > 0$.

Assumption 3.2 *There exists a sequence* $\{\mathfrak{F}_n\}_{n=1}^{\infty}$ *defined as above, with* $\lim_{n\to\infty} \frac{k_n}{n} = 0$, *and positive numbers* c_1, c_2 *such that* $\Pi(\mathfrak{F}_n^c) \leq c_1 e^{-nc_2}$ *for all but finitely many* n.

Assumption 3.2 means that the prior should put small mass on the set of functions that has too many jumps compared to the number of observations. As in [6], we can show that the denominator $m_n(y^n)$ is not exponentially small.

Lemma 3.3 *(Adapted from Lemma 4 in [6])*

Under assumption 3.1, for every $\epsilon > 0$,

$$P_0^{\infty}(y^{\infty} : m_n(y^n) \leq e^{-n\epsilon}, i.o.) = 0$$

In order to prove our main theorem, the following explicit bounds are needed:

Lemma 3.4 *Let* $f, g \in \mathfrak{F}_n$ *be step functions with at most* k_n *jumps, then*

$$\left| \int_0^1 |f(x) - g(x)|\, dx - \frac{\sum_i |f(x_i) - g(x_i)|}{n} \right| \leq \frac{4Kk_n}{n}$$
$$\left| \int_0^1 (f(x) - g(x))^2\, dx - \frac{\sum_i (f(x_i) - g(x_i))^2}{n} \right| \leq \frac{8K^2k_n}{n}$$

Proof. If $f - g$ has one or more jumps inside $[\frac{i}{n}, \frac{i+1}{n})$, which happens for at most $2k_n$ intervals, then $\left| \int_{i/n}^{i+1/n} |f(x) - g(x)|\, dx - \frac{|f(x_i)-g(x_i)|}{n} \right| \leq \frac{2K}{n}$, by the uniform boundedness of functions in $\mathfrak{F}_n$. If $f - g$ has no jumps inside $[\frac{i}{n}, \frac{i+1}{n})$, then $\left| \int_{i/n}^{i+1/n} |f(x) - g(x)|\, dx - \frac{|f(x_i)-g(x_i)|}{n} \right| = 0$. Sum up over i to get the first inequality, the second inequality is proved in the same way. □

With the above lemmas, we can derive our main theorem:

Theorem 3.5 *Under Assumptions 3.1 and 3.2, for every* $\epsilon > 0$

$$\lim_{n\to\infty} \Pi^n(A_\epsilon^c) = 0 \ a.s.\ P_0^{\infty}$$

Proof. Write

$$\Pi^n(A_\epsilon^c) = \Pi^n(A_\epsilon^c \cap \mathfrak{F}_n) + \Pi^n(A_\epsilon^c \cap \mathfrak{F}_n^c).$$

The second term on the right hand side goes to 0 by our assumption on the prior (the exact arguments of Lemma 5 in [6] can be followed even though y_i are not i.i.d.). So we only need to prove that the first term also goes to 0 a.s. P_0^∞. Similar to [6], let $C_n = A_\epsilon^c \cap \mathfrak{F}_n$. Now, write $\Pi^n(C_n) = \frac{m_n(y^n, C_n)}{m_n(y^n)}$. Let $r = r(n, \delta)$ be the covering number of $\mathfrak{F}_n$ in L_2 norm, and let $\{f_1, f_2, \ldots, f_r\}$ be the δ-net for $\mathfrak{F}_n$. Define

$$\tilde{E}_j = \{f \in \mathfrak{F}_n : \|f - f_j\|_{L_2} \leq \delta\}.$$

Let $E_1 = \tilde{E}_1$, and for $j > 1$ let $E_j = \tilde{E}_j \setminus \cup_{s=1}^{j-1} \tilde{E}_s$. Hence $\{E_1, \ldots, E_r\}$ are disjoint and cover C_n. We now write

$$\begin{aligned}
& m_n(y^n, C_n) \\
= & \sum_{j=1}^{r} m_n(y^n, C_n \cap E_j) \\
= & \sum_{j=1}^{r} \prod_{i=0}^{n} exp\{-\frac{(y_i - f_j(x_i))^2 - (y_i - f_0(x_i))^2}{2\sigma^2}\} \\
& \cdot \int_{C_n \cap E_j} \prod_{i=0}^{n} exp\{-\frac{(y_i - f(x_i))^2 - (y_i - f_j(x_i))^2}{2\sigma^2}\} \Pi(df)
\end{aligned}$$

where we pick f_j from the δ-net if $C_n \cap E_j \neq \emptyset$. The magnitude of the different terms in the last expression is estimated in Lemma 3.6,3.7 and 3.8. Combining these estimates with Lemma 3.3, and choosing appropriate value for δ as a function of ϵ, we get

$$\lim_{n \to \infty} \Pi^n(A_\epsilon^c) = 0 \ a.s. P_0^\infty$$

. $\square$

Lemma 3.6

$$\prod_{i=0}^{n} exp\{-\frac{(y_i - f_j(x_i))^2 - (y_i - f_0(x_i))^2}{2\sigma^2}\} \leq e^{-n\beta}$$

eventually, for $\beta = \frac{(\epsilon-\delta)^2}{16\sigma^2}$

Proof.

$$\begin{aligned}
& P^{\infty}(y^{\infty} : \prod_{i=0}^{n} exp\{-\frac{(y_i - f_j(x_i))^2 - (y_i - f_0(x_i))^2}{2\sigma^2}\} \geq e^{-n\beta}) \\
= \; & P^{\infty}(y^{\infty} : \prod_{i=0}^{n} exp\{-\frac{(y_i - f_j(x_i))^2 - (y_i - f_0(x_i))^2}{4\sigma^2}\} \geq e^{-n\frac{\beta}{2}}) \\
\leq \; & e^{n\frac{\beta}{2}} E \prod_{i} exp\{-\frac{(y_i - f_j(x_i))^2 - (y_i - f_0(x_i))^2}{4\sigma^2}\} \\
= \; & exp\{n(\frac{\beta}{2} - \frac{\sum_i (f_j(x_i) - f_0(x_i))^2}{8n\sigma^2})\} \\
\leq \; & exp\{n(\frac{\beta}{2} - \frac{1}{8\sigma^2}(\|f_j - f_0\|^2 - \frac{8K^2 k_n}{n}))\} \\
\leq \; & exp\{n(\frac{\beta}{2} - \frac{1}{8\sigma^2}((\epsilon - \delta)^2 - \frac{8K^2 k_n}{n}))\} \\
\leq \; & exp\{-n\frac{(\epsilon-\delta)^2}{32\sigma^2}\}
\end{aligned}$$

where the first inequality follows from Markov inequality, second inequality follows from Lemma 3.4, third inequality follows from the fact that if $f \in C_n \cap E_j \neq \emptyset$, then $\|f_j - f_0\| \geq \|f_0 - f\| - \|f - f_j\| \geq \epsilon - \delta$, the last inequality follows from k_n/n goes to 0, so $\frac{8K^2 k_n}{n}$ will be less than $\frac{(\epsilon-\delta)^2}{2}$ for n large enough. □

Lemma 3.7 *If* $\delta < 1$, *we have*

$$\int_{C_n \cap E_j} \prod_{i=0}^{n} exp\{-\frac{(y_i - f(x_i))^2 - (y_i - f_j(x_i))^2}{2\sigma^2}\}\Pi(df) \leq e^{n\gamma}$$

eventually, where $\gamma = \frac{2(4K+1)}{\sigma^2}\delta$

Proof. We write

$$\begin{aligned}
& E(\prod_{i=0}^{n} exp\{-\frac{(y_i - f(x_i))^2 - (y_i - f_j(x_i))^2}{2\sigma^2}\}) \\
= & \prod_i exp\{\frac{(f(x_i) - f_j(x_i))^2 - (f(x_i) - f_j(x_i))(2f_0(x_i) - f(x_i) - f_j(x_i))}{2\sigma^2}\} \\
\leq & exp\{\frac{n}{2\sigma^2}(\frac{\sum_i (f(x_i) - f_j(x_i))^2}{n} + \frac{4K\sum_i |f(x_i) - f_j(x_i)|}{n})\} \\
\leq & exp\{\frac{n}{2\sigma^2}(2\delta^2 + 4K \cdot 2\delta)\} \\
\leq & e^{n\frac{(4K+1)\delta}{\sigma^2}}
\end{aligned}$$

where the second inequality follows from Lemma 3.4 and that $||f - f_j||_{L_1} \leq ||f - f_j||_{L_2} \leq \delta$. Then,

$$\begin{aligned}
& P^{\infty}(\int_{C_n \cap E_j} \prod_{i=0}^{n} exp\{-\frac{(y_i - f(x_i))^2 - (y_i - f_j(x_i))^2}{2\sigma^2}\}\Pi(df) \geq e^{n\gamma}) \\
\leq & e^{-n\gamma} \int_{C_n \cap E_j} E\left(\prod_{i=0}^{n} exp\{-\frac{(y_i - f(x_i))^2 - (y_i - f_j(x_i))^2}{2\sigma^2}\}\right)\Pi(df) \\
\leq & e^{-n(\gamma - \frac{4K+1}{\sigma^2}\delta)} \\
= & e^{-n\frac{\gamma}{2}}
\end{aligned}$$

So we get the result after applying Borel-Cantelli lemma. □

Lemma 3.8 *For any $c' > 0$, we have $r(n, \delta) \leq e^{c'n}$ if $k_n \leq \alpha n$ for some α small enough ($\alpha = (\log_{\lceil \frac{2K}{\delta} \rceil} e)\frac{\delta^2 c'}{8K^2}$ suffices)*

Proof. The proof is just simple calculation. Choose a grid on the domain $[0, 1]$, and a grid on the range $[-K, K]$

$$\Delta_x = \{\frac{\delta^2}{8k_n K^2} \cdot i, i \in \mathbb{N}\} \cap [0, 1]$$

$$\Delta_y = \{\delta \cdot i, i \in \mathbb{N}\} \cap [-K, K]$$

Let $\tilde{\mathfrak{F}}_{\mathfrak{n}} = \{f \in \mathfrak{F}_{\mathfrak{n}}, f \text{ jumps only at points in } \Delta_x, \text{ and takes value in } \Delta_y\}$. Then it can be shown that $\tilde{\mathfrak{F}}_{\mathfrak{n}}$ is a δ-covering of $\mathfrak{F}_{\mathfrak{n}}$, and its size is at most $\lceil \frac{2K}{\delta} \rceil^{\frac{8k_n K^2}{\delta^2}}$. $\square$

As a corollary of Theorem 3.5, we have $||E(f|y_1, \ldots, y_n) - f_0||_{L_2} \to 0, a.s. P_0^\infty$, the proof is similar to Corollary 1 in [6].

Remark 3.9 *There is a general approach given in [39] that gives rates of convergence for non i.i.d. problems. The conditions involve constructing certain tests with exponentially small error. It is possible to adapt that general approach in our situation to get consistency result.*

3.1.3 Misspecification of the Prior

Sometimes the step functions are used as an approximation to a smoother function, the natural question is whether the posterior is still consistent in this case. We have the following result

Theorem 3.10 *Suppose the true function f_0 is contained in the set*

$$\mathfrak{G} = \{f : f \text{ is a continuous function on } [0,1], ||f||_\infty < K, ||f||_{Lip} < L\}$$

for some constant K and L, where $||f||_{Lip} := sup_{x \neq y \in [0,1]} |\frac{f(x)-f(y)}{x-y}|$ is the Lipschitz coefficient of f. We still have

$$\lim_{n \to \infty} \Pi^n(A_\epsilon^c) = 0 \ a.s. \ P_0^\infty$$

under Assumptions 3.1 and 3.2.

Proof. The proof is almost identical to that of Theorem 3.5, except that the explicit bounds proved in Lemma 3.4 (used in the proof of Lemma 3.6) cannot be used. Instead, we must prove a similar bound when $f \in \mathfrak{G}$ and $g \in \mathfrak{F}_{\mathfrak{n}}$, which we will do next. If $f - g$ has one or more jumps inside $[\frac{i}{n}, \frac{i+1}{n})$, which happens for at

most k_n intervals,then $\left|\int_{i/n}^{i+1/n} |f(x)-g(x)|\,dx - \frac{|f(x_i)-g(x_i)|}{n}\right| \leq \frac{2K}{n}$, by the uniform boundedness of functions in $\mathfrak{F}_n$ or $\mathfrak{G}$. If $f-g$ has no jumps inside $[\frac{i}{n}, \frac{i+1}{n})$, then $\left|\int_{i/n}^{i+1/n} |f(x)-g(x)|\,dx - \frac{|f(x_i)-g(x_i)|}{n}\right| \leq \frac{L}{2n^2}$. Summing up over i, we get

$$\left|\int_0^1 |f(x)-g(x)|\,dx - \frac{\sum_i |f(x_i)-g(x_i)|}{n}\right| \leq \frac{L}{2n} + \frac{2Kk_n}{n}$$

Similar bounds can be shown for $\left|\int_0^1 (f(x)-g(x))^2\,dx - \frac{\sum_i (f(x_i)-g(x_i))^2}{n}\right|$. □

We now present a construction of the prior that satisfies Assumptions 3.1 and 3.2. The construction is similar to the one used in [42]. A step function on $[0,1]$ is fully determined by a vector $\{k, s_1, s_2, \ldots, s_k, h_0, h_1, \ldots, h_k\}$, where k is the number of jumps, s_i is the location of the $i-th$ jump, and h_i is the value of f on $[s_i, s_{i+1})$ (with the understanding that $s_0 = 0, s_{k+1} = 1$). In the prior, we suppose that k is drawn from a Poisson distribution $Poiss(\lambda)$. Given k, $\{s_i\}_{i=1}^k$ is distributed as order statistics of k uniform random variables on $[0,1]$. Finally, $\{h_i\}_{i=0}^k$ are drawn independently from the uniform distribution on $[-K, K]$.

Assumption 3.2 is satisfied by the specification of the distribution on k.

Proposition 3.11 *Let $p(k)$ be the marginal distribution of Π on the number of jumps. If p is a Poisson distribution $Poiss(\lambda)$, then Assumption 3.2 is satisfied.*

Proof. We want to show that for appropriately chosen k_n,

$$\sum_{k=k_n}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} < c_1 e^{-c_2 n}$$

This can be satisfied by, say, $k_n = n/\log n$. The detailed calculation is omitted here. □

By simple calculation, if $p(k) \propto p^k$, i.e. $p(k)$ is a Geometric distribution, there is no sequence k_n that can satisfies $\lim \frac{k_n}{n} = 0$ and $\sum_{k=k_n}^{\infty} p(k) < c_1 e^{-c_2 n}$ simultaneously. However, by the inspection of the proofs, we can relax Assumption 3.2 somewhat.

Assumption 3.12 *For any $\alpha > 0$, there exists a sequence k_n, with $\frac{k_n}{n} \leq \alpha$, and positive numbers c_1, c_2 such that $\Pi(\mathfrak{F}_n^c) \leq c_1 e^{-nc_2}$ for all but finitely many n.*

The condition $\lim \frac{k_n}{n} = 0$ in Assumption 3.2 is only used in the proofs of Lemma 3.6, 3.7, 3.8. By inspection, the proof only requires that k_n/n is sufficiently small compared to ϵ. So Theorem 3.5 is still true when Assumption 3.2 is replaced by Assumption 3.12. Now for a geometrically distributed prior on k, $p(k) \propto p^k$, it satisfies Assumption 3.12 with $k_n = \alpha n$.

Proposition 3.13 *For the prior described above, Assumption 3.1 is also satisfied, whether $f_0 \in \mathfrak{F}$ or $f_0 \in \mathfrak{G}$*

Proof. First assume $f_0 \in \mathfrak{F}$ is a step function with k jumps, so f_0 can be equivalently specified as $\{k, s_1, s_2, \ldots, s_k, h_0, h_1, \ldots, h_k\}$. Define a L_2 neighborhood $M(f_0, \epsilon_1, \epsilon_2)$ of f_0 in $\mathfrak{F}$ as follows. Every step function f in $M(f_0, \epsilon_1, \epsilon_2)$ has the vector description $\{k, t_1, t_2, \ldots, t_k, g_0, g_1, \ldots, g_k\}$, such that $|t_i - s_i| < \epsilon_1, |g_i - h_i| < \epsilon_2$. For every such f, we have $\int_0^1 (f(x) - f_0(x))^2 \, dx < 4K^2 k\epsilon_1 + \epsilon_2^2$. By choosing ϵ_1, ϵ_2 small enough, we have $M(f_0, \epsilon_1, \epsilon_2) \subset A_\epsilon$. Since $p(k) > 0$ and the distributions on s_i and h_i are continuous, $\Pi(A_\epsilon) \geq \Pi(M(f_0, \epsilon_1, \epsilon_2)) > 0$.

If $f_0 \in \mathfrak{G}$, there is some k large enough and some step function g with k jumps such that $||f_0 - g||_2 < \epsilon/2$. Arguing as above for the step function g, we can construct some L_2 neighborhood of g, $M(g, \epsilon_1, \epsilon_2)$, such that $M(g, \epsilon_1, \epsilon_2) \subset \{f \in \mathfrak{F} : ||g - f||_{L_2} \leq \epsilon/2\}$. Then we have $M(g, \epsilon_1, \epsilon_2) \subset A_\epsilon$ by the triangular inequality, and $\Pi(A_\epsilon) \geq \Pi(M(f_0, \epsilon_1, \epsilon_2)) > 0$. □

3.1.4 Changing Priors

With simple modifications of the above assumptions, we can deal with the situation that the prior is dependent on n. From now on the prior is denoted by $\Pi_n(df)$ (note that Π^n with a superscript n is used to denote the posterior). In place of Assumption 3.1, we should impose the following assumption.

Assumption 3.14 *For any $\epsilon, t > 0$, there exists an M such that $\Pi_n(A_\epsilon) \geq e^{-tn}$ when $n > M$*

Under Assumption 3.14, we can prove the following (adapted from Lemma 1 in [79])

Lemma 3.15 *Under Assumption 3.14, for any $t > 0$*

$$P_0^n(y^n : m_n(y^n) \leq e^{-tn}) \to 0$$

We can then show the weak consistency (i.e. L_2 consistency in probability) of the posterior distribution under changing priors, whose proof is identical to the proof of theorem 3.5, using Lemma 3.15 in place of Lemma 3.3.

Theorem 3.16 *Under Assumptions 3.2 and 3.14, for any $\epsilon > 0$*

$$\Pi^n(A_\epsilon^c) \to 0, \textit{ in } P_0^n \textit{ probability}$$

As a special case, we can show the (weak) consistency of the posterior distribution when the number of jumps is a deterministic parameter rather than random. For that purpose, we use the following prior. The prior Π_n is supported on the set of step functions that have exactly k_n jumps. With reuse of notation, we let $\mathfrak{F}_\mathfrak{n}$ denote this set of functions (different from the definition of $\mathfrak{F}_\mathfrak{n}$ in section 3.1.2). The distributions on the jump locations s_i and function values h_i are the same as specified in section 3.1.3. Using this prior, we have

Proposition 3.17 *If $\lim_{n\to\infty} \frac{k_n}{n} = 0$ and $k_n \to \infty$, then the posterior is weakly consistent, whether the true function $f_0 \in \mathfrak{F}$ or $f_0 \in \mathfrak{G}$*

Proof. Assumption 3.2 is trivially satisfied since $\Pi_n(\mathfrak{F}_\mathfrak{n}^\mathfrak{c}) = 0$. We only need to verify Assumption 3.14. First assume $f_0 \in \mathfrak{F}$, i.e. f_0 is a step function with k jumps. When $k_n > k$, f_0 can also be thought of as step function with k_n jumps. From the proof of Proposition 3.17, we have $M(f_0, \epsilon_1, \epsilon_2) \subset A_\epsilon$ when $\epsilon_1 < \epsilon^2/8K^2k_n$ and $\epsilon_2 < \epsilon/2$.

By our construction of the prior, we have $\Pi_n(M(f_0, \epsilon_1, \epsilon_2)) = k_n!(2\epsilon_1)^{k_n}(\frac{\epsilon_2}{K})^{k_n}$, if $\epsilon_1 = \epsilon^2/Ck_n$ and $\epsilon_2 = \epsilon/C$ for some constant C large enough. By the assumption that $k_n/n \to 0$, we can show $k_n!(2\epsilon_1)^{k_n}(\frac{\epsilon_2}{K})^{k_n} = k_n!(\frac{2\epsilon^2}{Ck_n})^{k_n}(\frac{\epsilon}{CK})^{k_n} > e^{-tn}$ when n is large enough.

If $f_0 \in \mathfrak{G}$, we can use the same argument as in the proof of the second half of proposition 3.17. □

3.1.5 Conclusions

In this section we find sufficient conditions under which the Bayes estimate for a piecewise constant function is consistent in the L_2 topology. Even when the true function is not piecewise constant, the Bayesian procedure can produce consistent estimate. Under either a Poisson or a Geometric distribution on the number of change points, the posterior is consistent as soon as we put continuous prior distributions on the jump location and function value on each segment. Consistency of the posterior immediately implies consistency of the Bayes estimate, however, it does not guarantee that we can have a consistent estimate of the number of jumps. It seems that some additional conditions on the prior penalizing functions with a large number of jumps is probably required to prevent overfitting, which might be an interesting direction for further investigation.

3.2 Rates of Convergence of Posterior under Misspecification

3.2.1 Introduction

In [49], the authors consider the situation where one cannot expect to achieve consistency since the prior is misspecified. In this case, it is not surprising that the posterior will converge to the distribution in the support of the prior that is closest

to the true distribution measured in Kullback-Leibler divergence. Instead of using the usual entropy number or its local version, they used a new concept called covering number for testing under misspecification and studied rates by constructing a sequence of tests between the true distribution P_0 and another measure that is not necessarily a probability distribution. The new entropy number can be reduced to the usual entropy in the well-specified case. In this section, we study the posterior distribution also under the misspecified situation, without constructing a sequence of tests.

The goal of this section is two fold. First, we show that the approach in [90] [91] can be extended to the situation of misspecified prior rather straightforwardly, by introducing an α-entropy condition that is slightly stronger than that of [49]. Second, we show that using a more refined analysis, a result similar to Theorem 2.2 in [49] can be recovered. In particular, it shows that under the well-specified case, this approach indeed is more general than the approach of constructing a sequence of tests.

In 3.2.2, we introduce necessary notations and concepts and present the martingale construction due to [90]. In 3.2.3, we prove the main result and show that this approach is somehow more general than the one presented in [49]. We end this section with a discussion in 3.2.4.

3.2.2 Preliminaries

Let $\{X_1, X_2, \ldots\}$ be independent samples generated from distribution P_0, with corresponding lower case letter p_0 denoting the density with respect to some dominating measure μ. We are given a collection of distributions $\mathcal{P}$, and a prior Π on it with $\Pi(\mathcal{P}) = 1$. For simplicity, we assume that there exists a unique distribution $P^* \in \mathcal{P}$ that achieves minimum value of Kullback-Leibler divergence to the true distribution,

that is

$$E_0(\log \frac{p_0}{p^*}) \leq E_0(\log \frac{p_0}{p}), \text{ for all } p \in \mathcal{P}$$

where E_0 denotes the expectation under the true distribution P_0.

Let $R_n(p) = \prod_{i=1}^n p(X_i)/p^*(X_i)$, then the posterior mass for a set B is

$$\Pi^n(B) = \frac{\int_B R_n(p)\Pi(P)}{\int R_n(p)\Pi(P)} \tag{3.1}$$

Following [49], for $\epsilon > 0, 0 < \alpha < 1$ and some suitable semi-metric d on $\mathcal{P}$, we define the α-covering of the set $A = \{P \in \mathcal{P} : d(P, P^*) \geq \epsilon\}$ as a collection of convex sets $\{A_1, A_2, \ldots\}$ that covers A with the additional property that for any j,

$$\inf_{P \in A_j} -\log E_0(\frac{p}{p^*})^\alpha \geq \frac{\epsilon^2}{4} \tag{3.2}$$

and denote by $N_t(\epsilon, \alpha, A)$ the minimum integer N such that there exists $\{A_1, \ldots, A_N\}$ that forms such a cover, if N is finite.

This condition appears to be stronger than the concept of covering for testing under misspecification introduced in [49], which only requires that

$$\inf_{P \in A_j} \sup_{0<\alpha<1} -\log E_0(\frac{p}{p^*})^\alpha \geq \frac{\epsilon^2}{4} \tag{3.3}$$

In all the examples they gave in their paper, though, we can find a certain value of α only depending on the specification of the model that satisfies our condition. As shown in [49], when $\mathcal{P}$ is convex, we have $d^2(P, P^*) \leq -\log E_0(p/p^*)^{1/2}$ where d is a generalized Hellinger distance defined by $d^2(P_1, P_2) = \frac{1}{2}\int (p_1^{1/2} - p_2^{1/2})^2 p_0/p^* \, d\mu$, which reduces to the usual Hellinger distance in the well-specified case. In this situation, the 1/2-covering for testing can be replaced by the usual covering as shown in [49]. In general, allowing α to be different than 1/2 is required, since in the misspecified case, we cannot guarantee that $-\log E_0(p/p^*)^{1/2} > 0$, and we are obliged to choose

some smaller α in order to find the covering.

The predictive density constrained to a general set A is defined as

$$p_{nA}(x) = \int_A p(x)\Pi_A^n(P)$$

, where $\Pi_A^n(P) = 1_{\{P\in A\}}\Pi^n(P)/\Pi^n(A)$ is the posterior measure conditioned on A. The key identity noted by Walker (2003) is the following:

$$\int_A R_{n+1}(p)\Pi(P) = \frac{p_{nA}(X_{n+1})}{p^*(X_{n+1})}\int_A R_n(p)\Pi(P)$$

as can be verified easily. This in turn implies that

$$E_0[(\int_A R_{n+1}(p)\Pi(P))^\alpha|X_1,\ldots,X_n] = (\int_A R_n(p)\Pi(P))^\alpha E_0(\frac{p_{nA}}{p^*})^\alpha \tag{3.4}$$

which means that $\int_A R_n(p)\Pi(P)$ is a supermartingale when $E_0(p_{nA}/p^*)^\alpha < 1$.

3.2.3 Rates of Convergence

To study rates of convergence, for a sequence $\epsilon_n \to 0$, we let $A_n = \{P \in \mathcal{P} : d(P, P^*) \geq M\epsilon_n\}$, and let $A_{n,j}$ be an α-covering of A_n, i.e., $\{A_{n,j}\}$ are convex sets that covers A_n and

$$\inf_{P\in A_{n,j}} -\log E_0(\frac{p}{p^*})^\alpha \geq \frac{M^2\epsilon_n^2}{4}$$

Define

$$L_{k,j}^{(n)} = \int_{A_{n,j}} R_k(p)\Pi(P) \tag{3.5}$$

and

$$I_n = \int_{\mathcal{P}} R_n(p)\Pi(P)$$

To obtain a lower bound for I_n, which is the denominator in (3.1), we also need a condition on the prior mass for a Kullback-Leibler neighborhood of p^*, which is defined as

$$B(\epsilon, P^*; P_0) = \{P \in \mathcal{P} : -E_0(\log \frac{p}{p^*}) \leq \epsilon^2, E_0(\log \frac{p}{p^*})^2 \leq \epsilon^2\}$$

Theorem 3.18 *Assume that P^* is the unique minimizer in $\mathcal{P}$ of the Kullback-Leibler divergence to the true distribution with $E_0(\log(p_0/p^*)) < \infty$. For a sequence ϵ_n such that $\epsilon_n \to 0$ and $n\epsilon_n^2 \to \infty$, and $A_n, A_{n,j}$ defined as above. If the following conditions hold*

1) $e^{-n\epsilon_n K} \sum_j (A_{n,j})^\alpha$ for a sufficiently large constant K

2) $\Pi(B(\epsilon_n, P^*; P_0)) \geq e^{-Ln\epsilon_n^2}$ for a sufficiently large constant L

then $\Pi^n(P : d(P, P^*) \geq M\epsilon_n) \to 0$ in P_0^n probability.

Proof. First we observe that $P_{nA_{n,j}} \in A_{n,j}$ by the convexity of $A_{n,j}$. From the definition of α-covering, $\inf_{P \in A_{n,j}} -\log E_0(p/p^*)^\alpha \geq M^2\epsilon_n^2/4$, so the predictive density satisfies

$$E_0(\frac{p_{nA_{n,j}}}{p^*})^\alpha \leq e^{-M^2\epsilon_n^2/4}$$

Taking expectations in (3.4), with A replaced by $A_{n,j}$, we get

$$E_0(L_{k+1,j}^{(n)})^\alpha \leq E_0(L_{k,j}^{(n)})^\alpha e^{-M^2\epsilon_n^2/4}$$

and hence

$$E_0(L_{n,j}^{(n)})^\alpha \leq e^{-nM^2\epsilon_n^2/4}(\Pi(A_{n,j}))^\alpha$$

The posterior distribution can be bounded as follows:

$$\begin{aligned}\Pi^n(A_n) &\leq \sum_j \Pi^n(A_{n,j}) \\ &\leq \sum_j [\Pi^n(A_{n,j})]^\alpha = \sum_j \frac{(L_{n,j}^{(n)})^\alpha}{I_n^\alpha}\end{aligned}$$

Lemma 7.1 in [49] shows that when $n\epsilon_n^2 \to \infty$, for every $C > 0$, on a set Ω_n with probability converging to 1, we have $I_n \geq \Pi(B(\epsilon_n, P^*; P_0))e^{-n\epsilon_n^2(1+C)}$, so we can write

$$\begin{aligned}E_0(\Pi^n(A_n)) &= E_0(\Pi^n(A_n)1_{\Omega_n}) + E_0(\Pi^n(A_n)1_{\Omega_n^c}) \\ &\leq \frac{E_0 \sum_j (L_{n,j}^{(n)})^\alpha}{\Pi(B(\epsilon_n, P^*; P_0))^\alpha e^{-\alpha n\epsilon_n^2(1+C)}} + P_0(\Omega_n^c) \\ &\leq e^{-nM^2\epsilon_n^2/4 + \alpha n\epsilon_n^2(1+C) + \alpha n\epsilon_n^2 L} \sum_j \Pi(A_{n,j})^\alpha + P_0(\Omega_n^c)\end{aligned}$$

which converges to zero by condition 1) if M is sufficiently large. □

For a compact set of models $\mathcal{P}$, we can use the trivial bound

$$\sum_j \Pi(A_{n,j})^\alpha \leq N_t(\epsilon_n, \alpha, A_n)$$

, which gives the following result similar to Theorem 2.1 in [49], while they used a local version of the entropy instead.

Theorem 3.19 *If instead of condition 1) in Theorem 3.18, we assume $N_t(\epsilon_n, \alpha, A_n) \leq e^{n\epsilon_n^2}$, then for sufficiently large constant M,*

$$\Pi^n(P : d(P, P^*) \geq M\epsilon_n) \to 0 \text{ in probability.}$$

In order to get optimal rate for parametric models, [49] used a more refined assumption. In place of condition 2) in Theorem 3.18 above, they assumed

$$\frac{\Pi(P : J\epsilon_n < d(P, P^*) < 2J\epsilon_n^2)}{\Pi(B(\epsilon_n, P^*; P_0))} \leq e^{n\epsilon_n^2 J^2/8} \tag{3.6}$$

for all natural numbers n and J. In order to recover this result, we need a more careful analysis.

First, we define $A_n^J = \{P \in \mathcal{P} : M_n J\epsilon_n \leq d(P, P^*) < 2M_n J\epsilon_n\}$, with $\alpha-$covering $\{A_{n,j}^J\}$ defined similarly as before with the property: $\inf_{P\in A_{n,j}^J} -\log E_0(p/p^*)^\alpha \geq M_n^2 J^2\epsilon_n^2/4$. Let $\tilde{A}_{n,j}^J = A_{n,j}^J \cap A_n^J$, note that $\tilde{A}_{n,j}^J$ might not be convex even though $A_{n,j}^J$ is constrained to be so. Similarly, we can define $\tilde{L}_{k,j}^{(n),J}$ as in (3.5) with $A_{n,j}$ replaced by $\tilde{A}_{n,j}^J$. It is easy to see that the following still holds:

$$E_0(\tilde{L}_{k+1,j}^{(n),J})^\alpha = E_0(\tilde{L}_{k,j}^{(n),J})^\alpha E_0(\frac{p_{n\tilde{A}_{n,j}^J}}{p*})^\alpha \leq E_0(\tilde{L}_{k,j}^{(n),J})^\alpha e^{-M_n^2 J^2\epsilon_n^2/4}$$

even though $\tilde{A}_{n,j}^J$ might be nonconvex, since $P_{n\tilde{A}_{n,j}^J}$ is still contained in $A_{n,j}^J$ though not necessarily in $\tilde{A}_{n,j}^J$.

With A_n^J playing the role of A_n before, the same strategy in the proof of Theorem 3.18 can be followed to show that

$$\begin{aligned} E_0(\Pi^n(A_n^J)1_{\Omega_n}) &\leq \frac{E_0 \sum_j (\tilde{L}_{n,j}^{(n),J})^\alpha}{\Pi(B(\epsilon_n, P^*; P_0))^\alpha e^{-\alpha n\epsilon_n^2(1+C)}} \\ &\leq e^{-nM_n^2 J^2\epsilon_n^2/4+\alpha n\epsilon^2(1+C)} \frac{\sum_j \Pi(\tilde{A}_{n,j}^J)^\alpha}{\Pi(B(\epsilon_n, P^*; P_0))^\alpha} \end{aligned}$$

We will use the notation N_t^J to denote the α-covering number for A_n^J. We are now ready to prove the following:

Theorem 3.20 *Assume that P^* is the unique minimizer of the Kullback-Leibler divergence to the true distribution with $E_0(\log(p_0/p^*)) < \infty$. For a sequence ϵ_n such that $\epsilon_n \to 0$ and $n\epsilon_n^2$ bounded away from zero, and $A_n^J, \{A_{n,j}^J\}_{j=1}^{N_t^J}$ defined as above. If*

the following conditions hold

1) $N_t^J \leq e^{n\epsilon_n^2}$ *for all* $J \geq 1$

2) (3.6) is satisfied

Then we have

$$\Pi^n(P : d(P, P^*) \geq M_n\epsilon_n) \to 0$$

in probability for any sequence $M_n \to \infty$

Proof. We start by writing

$$\begin{aligned} E_0(\Pi^n(A_n)) &= \sum_{J=1}^{\infty} E_0(\Pi^n(A_n^J)) \\ &\leq \sum_J E_0(\Pi^n(A_n^J)1_{\Omega_n}) + P_0(\Omega_n^c) \\ &\leq \sum_J e^{-nM_n^2J^2\epsilon_n^2/4+\alpha n\epsilon_n^2(1+C)} \frac{\sum_j \Pi(\tilde{A}_{n,j}^J)^\alpha}{\Pi(B(\epsilon_n, P^*; P_0))^\alpha} + P_0(\Omega_n^c) \quad (3.7) \end{aligned}$$

we can bound the inner sum for each fixed J as

$$\sum_j \Pi(\tilde{A}_{n,j}^J)^\alpha \leq N_t^J \Pi(A_n^J)^\alpha \leq e^{n\epsilon_n^2}\Pi(A_n^J)^\alpha$$

since $\tilde{A}_{n,j}^J \subset A_n^J$ and using condition 1). Plugging this into (3.7) and using condition 2), we get

$$E_0(\Pi^n(A_n)) \leq \sum_{J\geq 1} e^{-n\epsilon_n^2M_n^2J^2/4+\alpha n\epsilon_n^2(1+C)+n\epsilon_n^2+\alpha n\epsilon_n^2M_n^2J^2/8} + P_0(\Omega_n^c)$$

By Lemma 7.1 of [49], $P_0(\Omega_n^c)$ can be made arbitrarily small by choosing C sufficiently large, under the condition that $n\epsilon_n^2$ is bounded away from zero. For any C, the sum above converges to zero since $M_n \to \infty$. □

3.2.4 Discussion

We demonstrated that rates of convergence of posterior distribution under misspecification can be established without construction of a sequence of tests. Theorem 3.20 we derived above is slightly weaker than Theorem 2.2 in [49] due to our use of assumption (3.2), which is stronger than (3.3). This said, we are not aware of any examples where the weaker condition (3.3) provides any advantage over (3.2). In [93], the authors demonstrated that using the martingale approach can improve on the rates slightly for some problems. Theorem 3.20 shows that the results by [49] is implied by our result, this is precisely true for well-specified problem, while for misspecified problem this is not conclusive due to the reason stated above. Unfortunately, we have not been able to construct an example that this approach provides a faster rate.

The extension to the case that the prior Π depends on n, and the case that there exists a finite number of points at minimal Kullback-Leibler divergence to the true distribution should be straightforward.

Chapter 4

Functional Data Analysis with Functional Repreducing Kernel Hilbert Spaces

An extension of reproducing kernel Hilbert space (RKHS) theory provides a new framework for modeling functional regression models with functional responses. The approach only presumes a general nonlinear regression structure as opposed to previously studied linear regression models. Generalized cross-validation (GCV) is proposed for automatic smoothing parameter estimation. The new RKHS estimate is applied to both simulated and real data as illustrations.

4.1 Introduction

In many experiments, functional data appear as the basic unit of observations. As a natural extension of the multivariate data analysis, functional data analysis provides valuable insights into these problems. Compared with the discrete multivariate analysis, functional analysis takes into account the smoothness of the high dimensional covariates, and often suggests new approaches to the problems that have not been discovered before. Even for nonfunctional data, the functional approach can often offer new perspectives on the old problem.

The literature contains an impressive range of functional analysis tools for various problems including exploratory functional principal component analysis, canonical correlation analysis, classification and regression. Two major approaches exist. The more traditional approach, carefully documented in the monograph [68], typically starts by representing functional data by an expansion with respect to a certain basis, and subsequent inferences are carried out on the coefficients. The most commonly utilized basis include B-spline basis for nonperiodic data and Fourier basis for periodic data. Another line of work by the French school, taking a nonparametric point of view, extends the traditional nonparametric techniques, most notably the kernel estimate, to the functional case. Some theoretical results are also obtained as a generalization of the convergence properties of the classical kernel estimate.

We are concerned with the regression problem in this chapter. In general, the

regression problem takes the form

$$y_i = F(x_i) + \epsilon_i \tag{4.1}$$

In traditional nonparametric inference, with x_i and y_i both being scalars, there exist a large number of methods including kernel and locally linear estimation. In functional data analysis, one or more of the components, x_i, y_i, and ϵ_i, are functions defined on some interval, here assumed to be the interval $[0, 1]$. Inferences are focused on the estimation of the structural component $F(x)$, with the residual ϵ modeling the noise or in general the component of observations not captured by the model.

In the case of scalar responses y, at least two nonparametric approaches have appeared in the literature. The first method uses a simple kernel regression estimate

$$\hat{F}(x) = \frac{\sum_i k(d(x_i, x))y_i}{\sum_i k(d(x_i, x))}$$

where d is a semi-metric on the space of functions. The second method is to use the RKHS framework, assuming the real valued function F is an element of the RKHS H, and the estimator is obtained as the minimizer of the regularized loss

$$\sum_i (y_i - F(x_i))^2 + \lambda ||F||_H^2 \tag{4.2}$$

The construction of the Hilbert space in this scalar response case involves no extra technical difficulties compared with the classical multivariate case as long as we have a metric on the space of functions x, and then the kernel can be constructed with $K(x_1, x_2) = k(d(x_1, x_2))$ for any positive definite function k. The representer theorem for RKHS implies that the solution to (4.2) has the form

$$F(x) = \sum_i \alpha_i K(x_i, x)$$

with real coefficients α_i. This representation can be plugged back into (4.2) and solve for the coefficients. Note that in both of these two nonparametric approaches, the same word "kernel" is used for different concepts, the exact meaning of the word should be clear from the context.

In the case of functional responses y, the classical parametric inferences assume that F is linear in x. More explicitly, the pointwise model assumes that

$$y(t) = \alpha(t) + \beta(t)x(t) + \epsilon(t)$$

while the integral model specifies that

$$y(t) = \alpha(t) + \int_0^1 \beta(s,t)x(s)\,ds + \epsilon(t) \tag{4.3}$$

In contrast to traditional linear regression models, now the coefficient β is a function on $[0,1]$, or a bivariate function on $[0,1] \times [0,1]$. To estimate the coefficient β, again a basis is chosen to represent the functions involved and the problem is reduced to a multiple linear regression model. To our knowledge, nonparametric approaches to functional analysis with functional responses has not been studied before and we will embark on this task in the current chapter.

We consider the functional response model (4.1) in which the structural component F is a mapping from some space of functions to another function space. We assume that $y(\cdot)$ belongs to a Hilbert space H. Although it is not necessary to assume that $x(\cdot)$ belong to the same space, or even that there is an inner product associated with it, it will be convenient to require that x is in the same space as y, as we will assume in the following.

In this chapter, we will develop an estimation procedure for functional response models within the framework of RKHS. For nonlinear models, the parametric approach above does not give satisfactory results. Our goal is to show that within the RKHS framework, a simple estimate can be developed which reduces to linear re-

gression computations very much like those derived in the parametric approach. Our work is motivated by the work of [66]. Unlike the case of scalar response models treated in [66], the extension we need is more complicated, as we will discuss in section 4.2. There we show how we should extend the notion and the construction of a RKHS in this new setting and also prove the corresponding representer theorem. In section 4.3, we present our new nonlinear model within the framework developed and also comment on the computations involved. Simulation studies and application to the well-known weather data are carried out in section 4.4. These results show clear advantage of our model compared to the parametric linear regression model in nonlinear contexts. A kernel estimate similar to that of [31] and [32] are also constructed for comparison. We conclude the chapter in section 4.5. Some technical details are deferred to the appendix.

4.2 Functional reproducing kernel Hilbert spaces

Following [89], a (real) RKHS H is a Hilbert space of real-valued functions defined on, say, the interval $[0,1]$, in which the point evaluation operator $L_t : H \to R, L_t(f) = f(t)$ is continuous. By Riesz representation theorem, this definition implies the existence of a bivariate function $K(s,t)$ such that

$$\begin{aligned} &K(s,\cdot) \in H, \text{ for all } s \in [0,1] \\ &\text{and (reproducing property)} \\ &\text{for every } f \in H \text{ and } t \in [0,1],\ \langle K(t,\cdot), f\rangle_H = f(t) \end{aligned} \tag{4.4}$$

The definition of a RKHS can actually start from a positive definite bivariate function $K(s,t)$ and RKHS is constructed as the completion of the linear span of $\{K(s,\cdot), s \in [0,1]\}$with inner product defined by $\langle K(s,\cdot), K(t,\cdot)\rangle_H = K(s,t)$.

In the regression model (4.1) with functional response, we are dealing with the

functional F taking values in the Hilbert space H. So the RKHS we construct should be a subset of $\{F : H \to H\}$. To define the RKHS in this case, we follow the same procedure as in [89].

Definition 4.1 *A (functional) RKHS $\mathcal{H}$ is a subset of $\{F : H \to H\}$. It is a Hilbert space, with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, in which the point evaluation operator is a bounded linear operator, i.e., $L_x : F \to F(x)$ is a bounded operator from $\mathcal{H}$ to H for any $x \in H$.*

The definition above is not useful for constructing a RKHS. For our purpose, we need to explicitly define the kernel associated with the space. The continuity of $F \to F(x)$ for any $x \in H$ is equivalent to the continuity of the mapping $F \to \langle F(x), g \rangle_H$ for any $x \in H$ and $g \in H$. By Riesz representation theorem applied to the Hilbert space $\mathcal{H}$, there exists an element K_x^g in $\mathcal{H}$ such that

$$\langle K_x^g, F \rangle_{\mathcal{H}} = \langle F(x), g \rangle_H \tag{4.5}$$

From the above, the mapping $g \to K_x^g$ is linear. By the boundedness of the point evaluation operator, we also have

$$\langle K_x^g, F \rangle_{\mathcal{H}} = \langle F(x), g \rangle_H \leq C||F||_{\mathcal{H}}||g||_H$$

which implies that the mapping $g \to K_x^g$ is also bounded. And so $g \to K_x^g(y)$ is a bounded linear operator for any $y \in H$, which we can denote by $K(x, y)$, i.e, $K(x, y)g := K_x^g(y)$, and we call $K(\cdot, \cdot)$ the reproducing kernel associated with $\mathcal{H}$. Note that in this case, the reproducing property is defined by $\langle K(x, \cdot)g, F \rangle_{\mathcal{H}} = \langle F(x), g \rangle_H$ for any $x \in H, g \in H$, this is just a rewriting of (4.5).

Two properties of the reproducing kernel are immediate. (a) $K(x, y) = K(y, x)^T$, where the superscript T denotes the adjoint operator. This is a simple consequence of the following sequence of identities making use of (4.5): $\langle K(x, y)g, f \rangle_H = \langle K_x^g(y), f \rangle_H =$

$\langle K_x^g, K_y^f \rangle_{\mathcal{H}} = \langle K_y^f(x), g \rangle_H = \langle g, K(y,x)f \rangle_H$. (b) $\sum_{i=1}^n \sum_{j=1}^n \langle K(x_i,x_j)f_i, f_j \rangle_H \geq 0$, which follows from $\sum_{i,j} \langle K(x_i,x_j)f_i, f_j \rangle_H = \sum_{i,j} \langle K_{x_i}^{f_i}, K_{x_j}^{f_j} \rangle_{\mathcal{H}} = || \sum_i K_{x_i}^{f_i} ||_{\mathcal{H}}^2$.

From the above discussions, we have the following definition of a positive definite kernel.

Definition 4.2 *A (functional) nonnegative definite kernel is a bivariate mapping on $H \times H$, taking values in $L(H)$, the space of bounded linear operators from H to itself, such that*

$$K(x,y) = K(y,x)^T$$

and

$$\sum_{i=1}^n \sum_{j=1}^n \langle K(x_i,x_j)f_i, f_j \rangle_H \geq 0$$

where $\{x_i\}$ and $\{f_i\}$ are any two sequences in H. If the double sum above is strictly positive whenever $\{x_i\}$ are distinct and $\{f_i\}$ are not all zero, K is a positive definite kernel.

Given a positive definite kernel defined as above, we can construct a unique RKHS of functions on H taking values also in H, with K as its reproducing kernel. The proof of this statement follows exactly the same lines as for the real-valued case.

4.3 Models for functional data

We consider the inference and prediction problem for model (4.1). Given the observed functional covariates and responses $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, a general approach to estimate F is to solve the following minimization problem:

$$\min_{F \in \mathcal{H}} \sum_{i=1}^n ||y_i - F(x_i)||_2^2 + \lambda ||F||_{\mathcal{H}}^2 \tag{4.6}$$

where a penalty term in $\mathcal{H}$-norm with smoothing parameter $\lambda > 0$ is added to the empirical risk as is usually done in the smoothing spline regression. We use the sim-

plest loss function $||x||_2^2 = \int_0^1 x^2(t)\,dt$ here, although other types of loss can certainly be considered.

The optimization problem above optimizes over the infinite-dimensional space $\mathcal{H}$ which is not feasible in general. Fortunately, the representer theorem below reduces this problem to finite dimensions (if you consider $\mathcal{H}$ as a vector space with elements in H acting as "constants"). The proof of the following is deferred to the appendix.

Theorem 4.3 *Given the observations $\{(x_i, y_i)\}_{i=1}^n$, the solution to (4.6) has the following representation*

$$\hat{F} = \sum_{i=1}^{n} K(x_i, \cdot)\alpha_i \tag{4.7}$$

with functional coefficients $\alpha_i \in H$.

Two difficulties arise at this stage. First, the construction of K in general is difficult and a search of the literature does not seem to provide us with any clues about how to construct a positive definite kernel in general. Second, even if the kernel is constructed, it is not clear how to solve (4.6) after we plug in the representation (4.7). Due to these difficulties, we are forced to choose the simplest functional kernel $K(x, y) = a(d(x, y))I$, where $a(\cdot)$ is a real positive definite function, $I \in L(H)$ is the identity operator, and d is a metric on H. We choose the simplest metric $d(x, y) = ||x - y||_2$ which is also used in [66]. It is unfortunate that we will not be able to provide more complicated examples of the kernel, but this estimate works reasonably well in all our experiments. It is clear that K defined in this way is nonnegative definite, but to prove that it is positive definite requires a little more extra work. We state this result formally as a theorem and its proof is to be found in the appendix.

Theorem 4.4 *The functional kernel $K(x, y) = a(||x - y||_2)I$ is positive definite if $a(||\cdot||_2)$ is a (real) positive definite function.*

After this dramatic simplification, we are able to solve problem (4.6). Let $a_{ij} = a(||x_i - x_j||_2)$ and make use of the representer theorem 1, we arrive at the following

optimization problem

$$\min_{\alpha_i} \sum_{i=1}^{n} ||y_i - \sum_j a_{ij}\alpha_j||_2^2 + \lambda \sum_{i,j} a_{ij}\langle \alpha_i, \alpha_j \rangle_H \tag{4.8}$$

From this point on, there are definitely more than one way to preceed. For example, one can represent each α_i by expansion with respect to a chosen basis. We choose instead to again compute (4.8) in the RKHS framework by assuming H is itself a (real) RKHS with reproducing kernel k. The loss term is first discretized, assuming the observations are made on a regular grid $\{t_1, \ldots, t_T\}$ on $[0, 1]$, then another application of the representer theorem in the real-value case stated in Theorem 3 below reveals the solution to be

$$\alpha_i = \sum_{l=1}^{T} b_l^i k(t_l, \cdot) \tag{4.9}$$

and we only need to compute the coefficients $\{b_l^i\}$. Formally, we have

Theorem 4.5 *Consider the raw data version of (4.8)*

$$\min_{\alpha_i} \sum_{i=1}^{n} \sum_{l=1}^{T} [y_i(t_l) - \sum_j a_{ij}\alpha_j(t_l)]^2 + \lambda \sum_{i,j} a_{ij}\langle \alpha_i, \alpha_j \rangle_H \tag{4.10}$$

The solution to (4.10) is of the form (4.9).

The readers are referred to the appendix for detailed formula involved in the computation of the solution to (4.10).

The model (4.10) looks similar to the classical smoothing spline estimation with two differences. First, in the first term of (4.10), instead of trying to smooth y by a single function, it tries to approximate each observation y_i as a combination of a common set of functions $\{\alpha_j\}_{j=1}^n$. The coefficients a_{ij} reflects the similarity of the covariates x_i and x_j. Second, an unconventional penalty term $\sum_{i,j} a_{ij}\langle \alpha_i, \alpha_j \rangle_H$ appears, which involves cross-over terms $\langle \alpha_i, \alpha_j \rangle_H$ with $i \neq j$. The loss term in (4.10)

seems natural and one could probably come up with this term without going through all the trouble of using functional RKHS and the derivation above. If this is the case, one would probably use a penalty term such as $\sum_i a_{ii}||\alpha_i||_H^2$ for regularization purposes. We call the solution to the problem (4.10) with this simpler penalty the modified RKHS estimate, as opposed to the RKHS estimate which solves the original problem (4.10).

In our implementation, we use the Gaussian kernel for both RKHSs, so $K(x,y) = exp\{-||x-y||_2^2/2\sigma^2\}I$ and $k(s,t) = exp\{-(s-t)^2/2\sigma'^2\}$. Thus there are at least three parameters, σ, σ', λ, that need to be specified. For the width parameters in the kernels, we simply choose σ to be the mean of all $||x_i - x_j||_2, i,j = 1,\ldots,n$, and σ' is similarly chosen to be the mean of $||t_i - t_j||_2, i,j = 1,\ldots,T$. These choices are of course suboptimal but produce good results in our experiences, so we do not bother to search over these parameter spaces. The choice of the smoothing parameter is arguably more important. Generalized cross-validation (GCV), which has been extensively studied in [89], can be used for the selection of λ. Given a grid of values for λ specified beforehand, GCV approximates the true error by

$$V(\lambda) = \frac{1}{n}||(I - A(\lambda))vec(Y)||_2^2/[\frac{1}{n}Tr(I - A(\lambda))]^2$$

where Y is the $n \times T$ matrix of raw observations $\{y_i(t_l)\}$ and $A(\lambda)$ is the "influence matrix" defined in Appendix B, and the final λ is chosen to be the one that minimizes $V(\lambda)$.

A nonparametric kernel estimate is studied in [31], [32], and [30], which is simply defined as

$$\hat{F}(x) = \frac{\sum_i k(||x_i - x||)y_i}{\sum_i k(||x_i - x||)} \tag{4.11}$$

In those papers, the authors only studied the kernel estimate for the model with scalar responses, but this estimate can obviously be used when the dependent variable is a curve. It also seems natural that we should smooth the response y_i before plugging

it into (4.11) if the observations are noisy. In the next section, this kernel estimate will be compared with our RKHS estimates and the linear parametric estimate.

4.4 Examples

4.4.1 Simulation study

One of the main goals of using RKHS estimate is to deal with nonlinearity in the data. In this simulation study, we compare four estimates. The RKHS estimate (4.7), its simpler version with modified penalty, the linear regression estimate (4.3) and the kernel estimate (4.11). The simulated data are generated by the following models:

1. $y(t) = \int_0^1 |t-s|x(s)\,ds$
2. $y(t) = \int_0^1 |t-s|x^2(s)\,ds$
3. $y(t) = \sin(2\pi t)x(t),\ t \in [0,1]$
4. $y(t) = \cos(\pi t)|x(t)|,\ t \in [0,1]$

Models (a) and (c) are linear in x while (b) and (d) are nonlinear. $x(\cdot)$ is generated as a standard Brownian motion with random starting position uniform in $[0,5]$, and $y(t)$ is computed using expressions (a)-(d). An equispaced grid of 50 points on $[0,1]$ is used. The raw observations for the dependent variable are the values of y on the grid points with i.i.d. standard normal noise added. The width parameters σ and σ' in RKHS and modified RKHS estimates are set to be the mean of the distances of the covariates from the training data as explained in the last section. After some experimentation, we use B-spline basis of order 4 with 10 equispaced knots in the linear regression estimate (4.3), with a penalty term involving the second partial derivatives of $\beta(s,t)$. The fitting of this linear model is performed using the fda package provided in R. A total of $n = 30$ replicates are used in model fitting. Since this is a simulation study, we generate a separate set of 50 replicates as validation to

Table 4.1: Results for the simulation study

	RKHS	Modified RKHS	Linear	Oracle Kernel	Kernel
Model (a)	1.000	0.944	0.327	0.910	2.773
Model (b)	1.000	1.000	11.632	1.984	2.035
Model (c)	1.000	0.992	0.471	1.643	1.954
Model (d)	1.000	1.007	2.892	1.459	1.735

choose the smoothing parameter λ in all three models, so GCV is not used here. For the kernel estimate, we again used a Gaussian kernel, and the only parameter is the width parameter inside the kernel. Although it can be fixed as in the RKHS estimator, and it indeed produces good results, we instead search over this one-dimensional space using the same validation data that was used to choose the smoothing parameter in other models. In kernel estimates, we use the raw data as y_i in (4.11). In simulation studies, we can also use $y(\cdot)$ in (4.11) before the standard normal noises are added, which we call the oracle kernel estimate for obvious reasons. In real applications, the performance of the kernel estimate should be worse than the oracle estimate if some kind of smoothing on the raw dependent data are use as a preprocessing step. The search over the width parameter put the kernel estimates in a favorable position compared to other estimates. Finally, after the parameters are estimated, we generate another 50 observations from model (a)-(d) as test data. These simulations are repeated 50 times for each model (a)-(d), and the mean of the mean square errors are reported in Table 4.1.

In Table 4.1, the errors for the RKHS estimates are normalized to be 1, and the errors of other estimates are shown as relative to the error for RKHS estimates. For linear models (a) and (c), the linear estimate clearly wins. For nonlinear models (b) and (d), the linear estimate performs badly compared to other estimates. The performance of the RKHS estimate and the modified RKHS estimate are almost identical. Although the kernel estimates do not perform as well as the RKHS estimates, it deserves further investigations due to its low computational complexity.

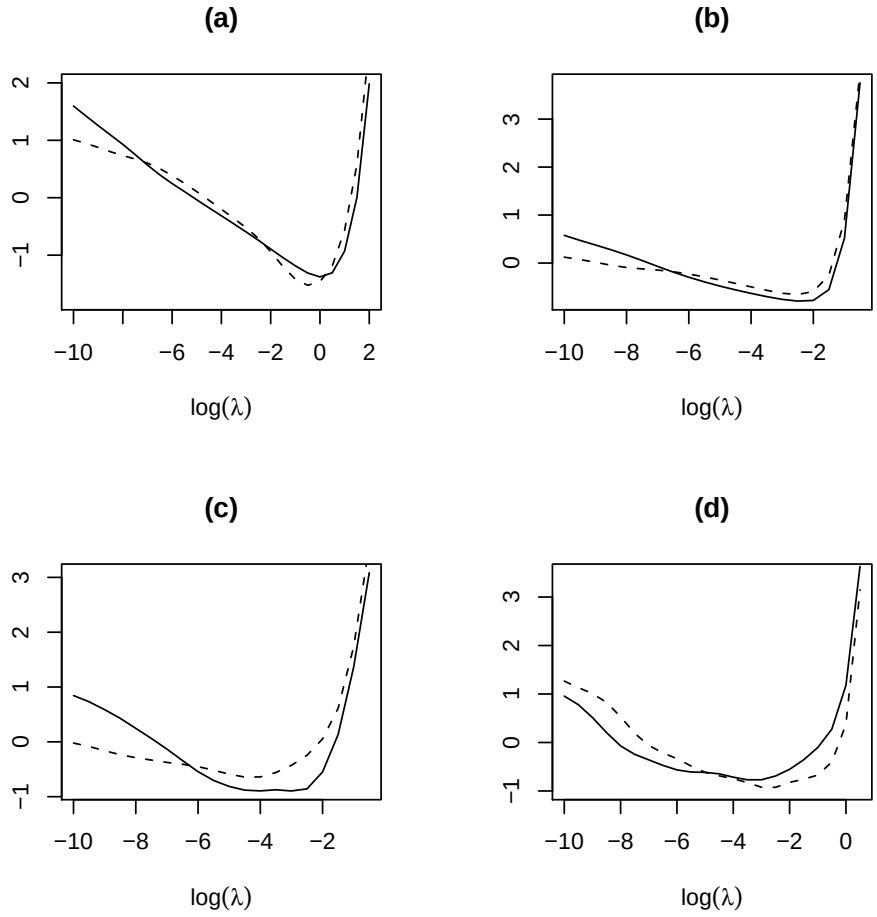


Figure 4.1: Comparison of the GCV with the error computed from validation data. The solid curves are the GCV estimates, the dashed curves are the error computed from validation set. The curves are shifted and normalized to show the shape of the curves rather than its absolute magnitudes.

Next we study the performance of GCV for estimating the smoothing parameter in the RKHS estimate and compare it to the estimate obtained from the validation data, which act as the background truth in our simulation study. Figure 4.1 show that for all four models (a)-(d), the GCV correctly identifies a good smoothing parameter to use in these cases.

4.4.2 Application to the weather data

The daily weather data consists of daily temperature and precipitation measurements recorded in 35 Canadian weather stations. These data are plotted in Figure 4.2. To save on computations, we subsample the data and use only the weekly measurements, so each observation consists of functional data observed on a equispaced grid of 53 points. We treat the temperature as the independent variable and the goal is to predict the corresponding precipitation curve given the temperature measurements. As is previously done, we set the dependent variable to be the log tranformed precipitation measurements, and a small positive number is added to the values with 0 precipitation recorded. The prediction of our RKHS estimate is shown in Figure 4.3 for four weather stations. Each plot is produced by fitting the model on the other 34 replicates, using GCV to choose the smoothing parameter, and then finally calculating the predicted precipitation based on the temperature curve. The figure shows reasonable prediction accuracy and can be compared to the results presented in Chapter 16 of [68].

4.5 Conclusions

We proposed a new approach to fitting a nonlinear functional regression model for functional responses. The simulations we conducted demonstrated the clear advantage of the RKHS estimate over the linear regression model when the true model is nonlinear. The estimate is also better than the simplistic kernel estimates used in

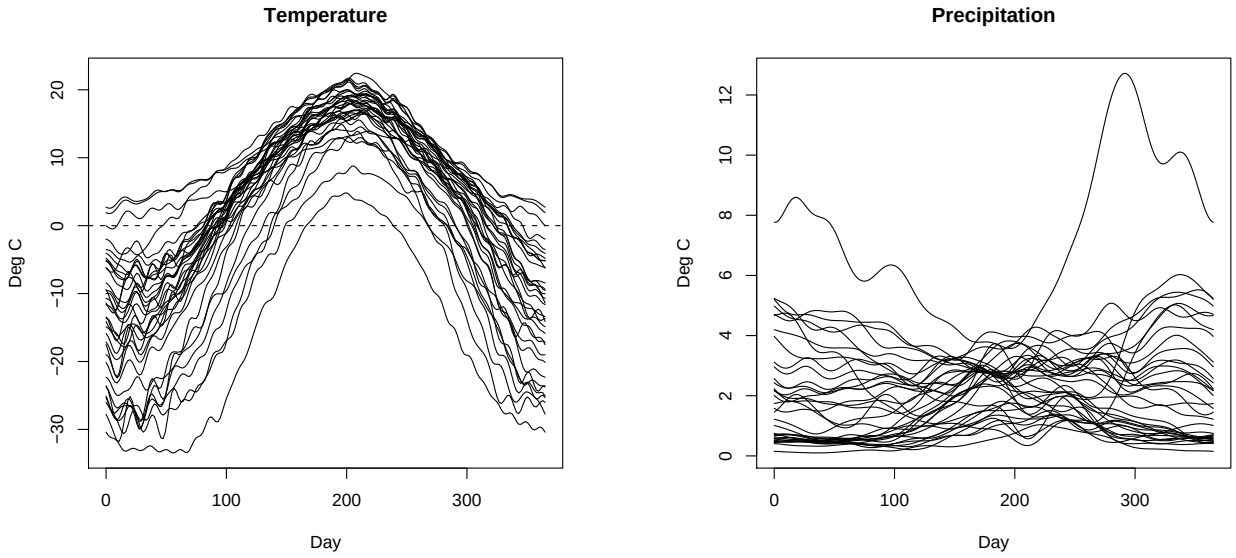


Figure 4.2: Daily weather data for 35 Canadian stations, the curves plotted here result from using smoothing splines to fit the raw data.

traditional nonparametric regression.

The advantages of the RKHS estimates are tied with the additional computational costs. In problem (4.10), we are optimizing over the same number of curves as there are the number of replicates which is computationally difficult when n is large. Approximate solutions such as choosing a limited number of kernel basis centered around selected covariates may prove to be useful.

Appendix A Proofs for Theorem 1-3

Proof of Theorem 1. Denote by $\mathcal{H}_0$ the subspace of $\mathcal{H}$ spanned by the kernel centered at the observed covariates: $\mathcal{H}_0 = \{\sum_{i=1}^n K(x_i, \cdot)\alpha_i, \alpha_i \in H\}$. Any $F \in \mathcal{H}$ can be written as $F = F_0 + G$, where $F_0 \in \mathcal{H}_0$ and $G \perp \mathcal{H}_0$ in $\mathcal{H}$. Then for any $j \in \{1, \ldots, n\}$, and any $h \in H$, $\langle G(x_j), h\rangle_H = \langle K(x_j, \cdot)h, G\rangle_{\mathcal{H}} = 0$ by the reproducing property (4.5), this implies $G(x_j) = 0$ by the arbitrariness of $h \in H$. Also, by the orthogonality of

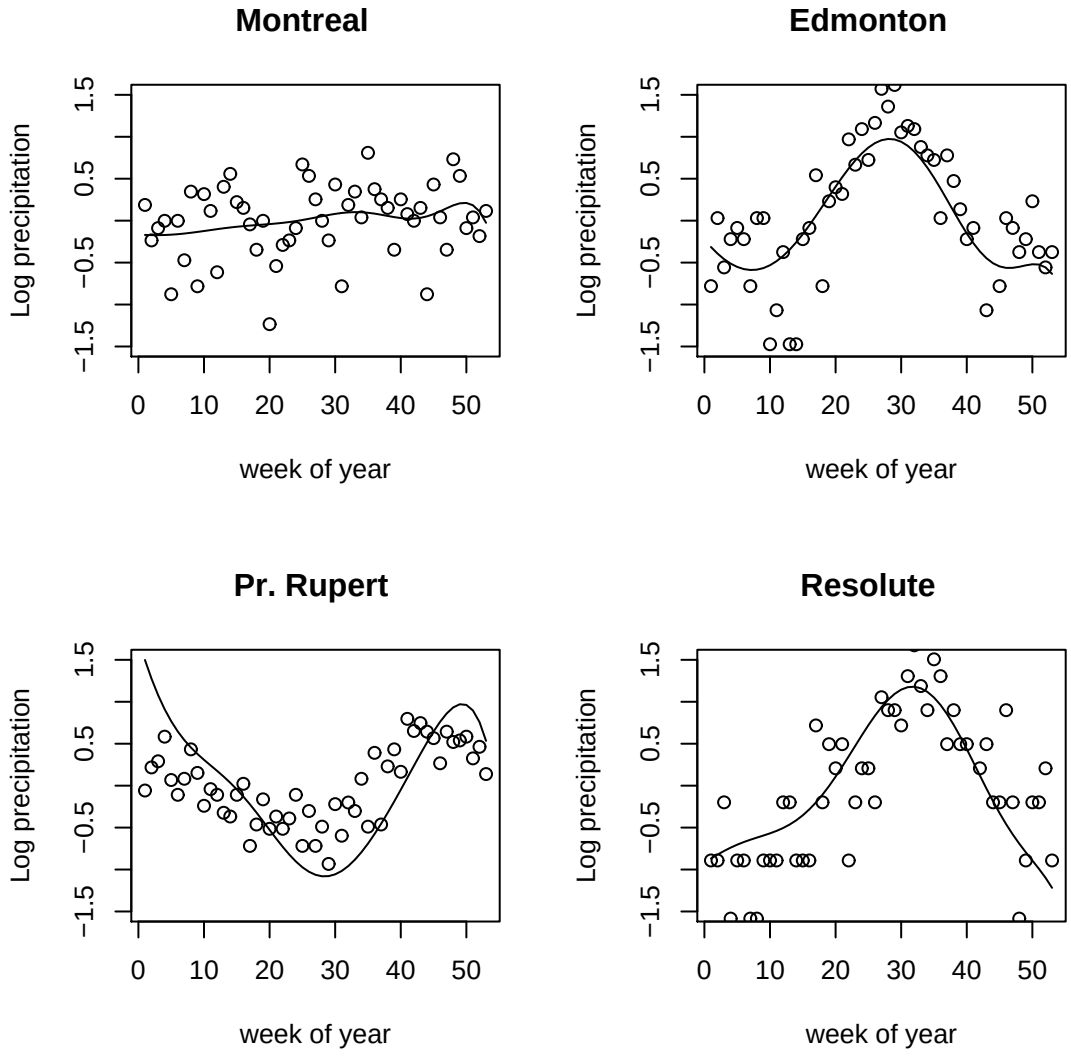


Figure 4.3: Raw data (points) and predictions (solid) of log precipitation for four weather stations.

G and F_0, $||F||^2_{\mathcal{H}} = ||F_0||^2_{\mathcal{H}} + ||G||^2_{\mathcal{H}} > ||F_0||^2_{\mathcal{H}}$ if G is nonzero. This implies that

$$\sum_{i=1}^{n} ||y_i - F(x_i)||_2^2 + \lambda ||F||^2_{\mathcal{H}} > \sum_{i=1}^{n} ||y_i - F_0(x_i)||_2^2 + \lambda ||F_0||^2_{\mathcal{H}}$$

if G is nonzero. So the minimizer $\hat{F}$ belong to $\mathcal{H}_0$

Proof of Theorem 2. We want to first show that $\sum_{i,j} \langle K(x_i, x_j) f_i, f_j \rangle_H \geq 0$. Since $\langle K(x_i, x_j) f_i, f_j \rangle_H = a_{ij} \langle f_i, f_j \rangle_H$, the nonnegativity follows immediately from Schur's Lemma, which asserts that the Hadamard product of two nonnegative matrices is again a nonnegative matrix.

If $\sum_{i,j} \langle K(x_i, x_j) f_i, f_j \rangle_H = 0$, and $\{x_i\}$ are distinct. Let $A = \{a_{ij}\}, B = \{\langle f_i, f_j \rangle_H\}$. A is positive definite and B is nonnegative definite. We have the factorization $B = E^T E$. So $\sum_{i,j} \langle K(x_i, x_j) f_i, f_j \rangle_H = \sum_{i,j} A_{ij} B_{ij} = \sum_{i,j} A_{ij} (E^T E)_{ij} = \sum_{i,j,k} A_{ij} E_{ki} E_{kj} = \sum_k (\sum_{i,j} E_{ki} A_{ij} E_{kj})$. By the positive definiteness of A, we get $E = 0$, which in turn implies $f_i = 0$ for all i.

Proof of Theorem 3. This is similar to the classical proof with smoothing splines. We write $\alpha_i = \alpha_{i,0} + g_i$, with $\alpha_{i,0} \in span\{k(t_l, \cdot)\}$ and g_i in its orthogonal complement. Similar to the proof of theorem 1, we only need to show that $\sum_{i,j} a_{ij} \langle \alpha_{i,0} + g_i, \alpha_{j,0} + g_j \rangle_H \geq \sum_{i,j} a_{ij} \langle \alpha_{i,0}, \alpha_{j,0} \rangle_H$ with equality only if $g_i = 0$ for all i. The proof of this statement is contained in the proof of Theorem 2.

Appendix B Computational details

We detail the computations involved in solving the model (4.10). The calculations are very similar to those used in linear regression model (4.3) using basis expansion, which is not surprising due to the representer theorem. Let $A = \{a_{ij}\}$ and B be the n by T matrix containing the coefficients in the expansion (4.9): $B = \{b_l^i\}$. Also, denote the $n \times T$ matrix $\{y_i(t_l)\}$ by Y and $T \times T$ matrix $\{k(t_i, t_j)\}$ by K. The objective

function (4.10) can be written in matrix form:

$$Tr((Y - ABK)(Y - ABK)^T) + \lambda Tr(ABKB^T)$$

Taking the derivative with respect to B, we want to solve

$$(A^T A)B(KK^T) + \lambda ABK = AYK$$

vectorizing the above equation gives us a system of linear equations

$$[(KK^T) \otimes (A^T A) + \lambda K \otimes A]vec(B) = (K \otimes A)vec(Y)$$

So we have the formula

$$vec(B) = [K \otimes A + \lambda I]^{-1} vec(Y)$$

and the fitted values are $\hat{Y} = ABK$, the vectorized version of this is

$$vec(\hat{Y}) = (K \otimes A)[K \otimes A + \lambda I]^{-1} vec(Y)$$

the matrix $A(\lambda) := (K \otimes A)[K \otimes A + \lambda I]^{-1}$ is the influence matrix used in the calculation of GCV.

Chapter 5

Conclusions

In this thesis, we first developed a new model for the general sequence data that are produced by the recent new technology in biology. Chapters 1 focus on building a new HMM-like model to model the variation in the dye intensities for the (transformed) microarray data. This new model gives us great flexibility in fitting more complex data. We gave efficient algorithms for the inference procedure and a principled approach to parameter estimation and model selection. The analysis on real data showed that the new model fit the sequence much better than the traditional model. We also presented several possible extensions, which added to the power of our model. We expect that the model is more generally applicable than the one biological example provided, and will continue to explore its versatility.

In future work, we hope to apply the model on multiple arrays as explain in section 1.4.5 when replicated data become available. With multiple arrays, which in effect reduce the annealing noise, we can hope to get better prediction for the binding sites. Of course, more experimentally validated data will be required to give us confidence on the results reported. We have not used the fully Bayesian model selection in our current implementation, it would be interesting to see how it works compared to the model selection conditioned on the hyperparameters. Other possibilities like MCMC can be explored. Although for our application in Chapter 1, this probably would not differ much because of the large amount of data we have, it might be interesting for other applications and as a general statistical question. Also, we have not looked into the detail of how the penalized regression used in parameter estimation would affect our inference. Simulations can be done to estimate the bias caused by the penalty imposed. All those questions deserve further investigation.

In Chapter 2, we proposed a new algorithm for image superresolution. This new approach can successfully address the problem of oversmoothing in many other approaches. It remains to explore other possible applications of this variational models.

In Chapter 3, we obtained some results on the asymptotics of the posterior distribution. One possible future direction is the study of the model selection properties of

Bayesian models. As far as we know, all current studies only focus on the estimation consistency of the posterior distribution. For example, in parametric linear regression problems, the estimation consistency ensures the prediction accuracy of the model, but does not imply that it put enough mass on the corrected model. Model selection is important especially in applications with high-dimensional covariates. We will try to address model consistency of the posterior distribution in the future.

In Chapter 4, we extended the theory of RKHS to deal with Hilbert space-valued functions. The framework makes it possible to study nonlinearity of a functional model with functional response which is not previously possible. Preliminary result shows the advantages of this new approach in the simulation studies. One problem of this model is that when the new observed functional covariate is far away from the training data, the prediction error will be unacceptably large. A combination of linear and nonlinear models might prove to be useful in this situation as a generalization of partially linear models in classical regression. Also, a test of nonlinearity should be developed to justify its usage. Also, asymptotic theory for the model will be valuable. The result in the real-valued case might be extensible to our case using the theory of empirical processes in Banach space.

Bibliography

[1] H. Akaike, Information theory and an extension of the maximum likelihood principle. *2nd International Symposium on Information Theory*, 267-281, 1973

[2] D. Aldous, Probability approximations via the Poisson clumping heuristic, *Springer*, 1989

[3] H. Attias, A variational Bayesian framework for graphical model, *NIPS*, 2000

[4] T.L. Bailey and C. Elkan, The value of prior knowledge in discovering motifs with MEME, *Proc. Third Conf Intell Syst Mol Biol* 3:21-29, 1995

[5] A. Barron, J. Rissanen and B. Yu, The minimum description length principle in coding and modeling, *IEEE Trans. Information Theory*, 44(6):2743-2760, 1998

[6] A. Barron, M.J. Schervish and L. Wasserman, The consistency of posterior distributions in nonparametric problems. *Ann. Statist.* 27, 536-561, 1999

[7] F. Bartolucci, L. Scaccia and A. Mira, Efficient Bayes factor estimation from the reversible jump output, *Biometrika*, 93(1):41-52, 2006

[8] J.O. Berger,Statistical decision theory and Bayesian analysis, *Springer*, 1985

[9] D. Blackwell and L. Dubins, Merging of opinions with increasing information. *Ann. Math. Statist.* 33, 882-886, 1962

[10] J.V. Braun and H. Müller, Statistical methods for DNA sequence segmentation, *Statistical Science,* 13(2):142-162, 1998

[11] P.J. Brockwell, and R.A. Davis, *Time series: theory and methods*, *2nd ed, Springer*, 1987

[12] T. Caelli and T. Caitano, Graphical models for graph matching: approximate models and optimal algorithms, *Pattern Recognition Letters,* 26:339-346, 2005

[13] B.P. carlin and T.A. Louis, Bayes and empirical Bayes methods for data analysis, *Chapman and Hall*, London, 1996.

[14] T. Chan, S. Osher and J. Shen, The digital TV filter and nonlinear denoising, *IEEE Transactions on Image Processing,* 10(2), 231-241, 2001

[15] T. Chan and J. Shen, Mathematical models for local nontexture inpaintings, *SIAM Journal on Applied Mathematics,* 62(3), 1019-1043, 2002

[16] E. Charniak, Statistical language learning, *MIT Press*, 1993

[17] W. Chu, Z. Ghahramani, D. Wild, A graphical model for protein secondary structure prediction, *ICML*, 2004

[18] G. Ciuperca, A. Ridolfi and J. Idier, Penalized maximum likelihood estimator for normal mixtures, *Scandinavian Journal of Statistics,* 30(1):45-59, 2003

[19] M. Clyde, Bayesian model averaging and model search strategies, In *Bayesian Statistics 6*, 157-185, Valencia, Spain, 1999

[20] A. Corduneanu and J.C. Platt, Learning spatially-variable filters for super-resolution of text, *IEEE International Conference on Image Processing,* 1, 849-852, 2005

[21] A. Dempster, N. Laird, and D. Rubin, Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistical Society(B)*, 39(1):1-38, 1977.

[22] P. Diaconis and D. Freeman, On the consistency of Bayes estimates (with discussion). *Ann. Statist.* 14, 1-67, 1986

[23] Y. Ding, C.Y. Chan, C.E. Lawrence, RNA secondary structure prediction by centroids in a Boltzmann Weighted Ensemble, *RNA* 11(8):1157-1166, 2005

[24] Y. Ding, C. Chan and C.E. Lawrence, Clustering of RNA secondary structures with applications to messenger RNAs. *Journal of Molecular Biology (In press)*

[25] R. Douc, A. Guillin, J.M. Marin and C.P. Robert, Convergence of adaptive mixtures of importance sampling schemes, *Annals of Statistics, to Appear*

[26] S.R. Eddy, Profile hidden Markov models, *Bioinformatics*, 14(9):755-763, 1998

[27] J. Fan and I. Gijbels, Local polynomial modeling and its application. *Chapman and Hall*, 1996

[28] J. Fan and Q. Yao, Nonlinear time series: nonparametric and parametric methods, *Springer*, 2003

[29] J.D. Ferguson, Variable duration models for speech, in *Symp. Application of Hidden Markov Models to Texts and Speech,* 143-179, 1980

[30] F. Ferraty, A. Mas and P. Vieu, Nonparametric regression from functional data: inference and practical aspects. *Australian and New Zealand Journal of Statistics*, to appear, 2007

[31] F. Ferraty and P. Vieu, Curves Discrimination: a Nonparametric Functional Approach. *Computational Statistics and Data Analysis*, 44, 161-173, 2003

[32] F. Ferraty and P. Vieu, Nonparametric models for functional data, with applications in regression, time series prediction and curves discrimination. *Journal of Nonparametric Statistics*, 16, 111-125, 2004

[33] W.T. Freeman, T.R. Jones and E.C. Pasztor, Example-based super-resolution, *IEEE Computer Graphics and Applications,* 22(2), 56-65, 2002.

[34] A. Gelman, J.B. Carlin, H.S. Stern and D.B. Rubin, Bayesian data analysis, *Chapman and Hall*, 1995

[35] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE-PAMI*, 6:721-741, 1984

[36] H.Kunsch, S. Geman and A.Kehagias, Hidden Markov random fields, *Annals of Applied Probability,* 5:577-602, 1995

[37] Z. Ghahramani and M.I. Jordan, Factorial hidden Markov models, *Machine Learning*, 29:245-273, 1997

[38] S. Ghosal, J.K. Ghosh and A.W. van der Vaart, Convergence rates of posterior distributions. *Ann. Statist.* 28, 500-531, 2000

[39] S. Ghosal and A.W. van der Vaart, Convergence rates of posterior distributions for non i.i.d. observations. *Ann. Statist. to appear*, 2007

[40] A.G. Glen, L.M. Leemis and J.H. Drew, Computing the distribution of the product of two continuous random variables, *Computational Statistics and Data Analysis*, 2004

[41] O. Gotoh, An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705-708, 1982

[42] P. Green, Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination, *Biometrika*, 82(4):711-732, 1995

[43] H. He and L.P. Kondi, A regularization framework for joint blur estimation and super-resolution of video sequences, *IEEE International Conference on Image Processing,* 3, 329-332, 2005

[44] J.A. Hoeting, D. Madigan, A.E. Raftery and C.T. Volinsky, Bayesian model averaging: A tutorial, *Statistical Science*, 44(4):382-417, 1999

[45] H. Huang, M.J. Kao, X. Zhou, J. Liu and W. Wong, Patters in Markov sequences with application to promoter element identification, *J. Computational Biology*, 11(1):1-14

[46] C. Ji and L. Seymour, A consistent model selection procedure for Markov random fields based on penalized pseudolikelihood, *Ann. Appl. Probab.*, 6(2):423-443, 1996

[47] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola and L.K. Saul, An introduction to variational methods for graphical models, In *Learning in Graphical Models*, Cambridge, MIT Press, 1999

[48] S. Karlin and S.F. Altschul, Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes, *PNAS*, 87:2264-2268

[49] B.J.K. Kleijn and A.W. van der Vaart, Misspecification in infinite-dimensional Bayesian statistics. *Ann. Statist.* 34, 837-877, 2006

[50] W. Li, C.A. Meyer and X. Liu, A hidden Markov model for analyzing chIP-chip experiments on genome tiling arrays and its application to p53 binding sequences. *Bioinformatics* 21:i274-i282, 2005

[51] X. Li and W. Wong, Sampling motifs on phylogenetic trees, *PNAS* 102(27):9481-9486 , 2005

[52] R. Little and D. Rubin, Statistical analysis with missing data, *Wiley*, 2002

[53] J.S. Liu and R. Chen, Sequential Monte Carlo methods for dynamic systems, *JASA*, 93(443):1032:1044, 1998

[54] J.S. Liu and C.E. Lawrence, Bayesian inference on biopolymer models, *Bioinformatics,* 15(1):38-52, 1999

[55] J.S. Liu, Monte Carlo strategies in scientific computing, *Springer*, 2002.

[56] S.E. Levinson, Continuously variable duration models and selected application in speech recognition, *Comput. Speech Lang.*, 1(1):29-45, 1986

[57] P.T. Lowary and J. Widom, Nucleosome packaging and nucleosome positioning of genomic DNA, *PNAS*, 94:1183-1188, 1997

[58] C. Mitchell, M. Harpert and L. Jamieson, On the complexity of explicit duration HMM's, *IEEE Tran. Speech Audio Processing*, 2:213-217, 1995

[59] N. Murata, S. Yoshizawa, S. Amari, Network Information Criterion: determining the number of hidden units for an artificial neural network model, *IEEE Trans. Neural Networks,* 5(6):865-872, 1994

[60] S.B. Needleman and Ch.D.Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins.*Journal of Molecular Biology*, 48:443-453, 1970

[61] A. Nemirovski, Topics in non-parametric statistics, in M. Emery, A.Nemirovski, D.Voiculescu, *Lecture Notes in Mathematics 1738*, Springer, 2000

[62] M.A. Newton and Y. Zhang, A recursive algorithm for nonparametric analysis with missing data, *Biometrika*, 86:15-26, 1999

[63] M.A. Newton, C.M. Kendziorski, C.S. Richmond, F.R. Blattner and K.W. Tsui, On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data, *Journal of Computational Biology*, 8(1):37-52, 2001

[64] D. Oakes, Direct calculation of the information matrix via the EM algorithm, *Journal of the Royal Statistical Society(B)*, 61(2):479-482, 1999

[65] L. Pachter, M. Alexandersson and S. Cawley, Applications of generalized pair hidden Markov models to alignment and gene finding problems, *Proceedings of the Fifth Annual International Conference on Computational Biology,* 241-248, 2001

[66] C. Preda, Regression models for functional data by reproducing kernel Hilbert spaces methods. *Journal of Statistical Planning and Inference*, 137, 829-840, 2007

[67] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE,* 77(2):257-286, 1989

[68] J.O. Ramsay and B.W. Silverman, *Functional Data Analysis*, 2nd edn. New York: Springer, 2005

[69] B. Ren et.al., Genome-wide location and function of DNA binding Proteins, *Science*, 290:2306-2309, 2000

[70] S.I. Resnick, Adventures in stochastic processes, *Birkäuser*, 1992

[71] S.T. Roweis and L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science,* 290, 2000

[72] L.I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D,* 60, 259-268, 1992

[73] P.J. Sabo et.al. A high-resolution map of human chromatin accessibility in vivo, *to appear*

[74] R.R. Schultz and R.L. Stevenson, Extraction of high-resolution frames from video sequences, *IEEE Transactions on Image Processing,* 4(6), 996-1011, 1996

[75] L. Schwartz, On Bayes Procedures. *Z. Wahrsch. Verw. Gabiete* 4, 10-26, 1965

[76] G. Schwarz, Estimating the dimension of a model, *Annals of Statistics,* 6(2):461-464, 1978

[77] J. Shao, Linear model selection by cross-validation, *JASA*, 88(422):486-494, 1993

[78] J. Shao, An asymptotic theory for linear model selection, *Statistica Sinica,* 7:221-261, 1997

[79] X. Shen and L. Wasserman, Rates of convergence of posterior distributions. *Ann. Statist.* 29, 79-100, 2001

[80] D. Stanford and A. Raftery, Approximate Bayes factors for image segmentation: the pseudolikelihood information criterion (PLIC), *IEEE-PAMI* 24(11):1517-1520, 2002

[81] M. Stone, An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, *Journal of the Royal Statistical Society.(B)*, 39(1):44-47, 1977

[82] J.D. Storey, The positive false discovery rate: a Bayesian interpretation and the q-value, *Annals of Statistics,* 31(6):2013-2035, 2003

[83] W. Thompson, E.C. Rouchka, C.E. Lawrence, Gibbs recursive sampler: finding transcription factor binding sites, *Nucleic Acids Res*, 31:3580-3585, 2003

[84] A.P. Varga and R. Moore, Hidden Markov model decomposition of speech and noise", *Proc. IEEE Conf. on Acoustics Speech and Sig. Proc. (ICASSP)*, 1990.

[85] M.J. Wainwright and M.I. Jordan, Graphical models, exponential families, and variational inference, *Technical Report 649*, Department of Statistics, University of California, Berkeley, 2003.

[86] K. Watanabe and S. Watanabe, Stochastic complexity of Gaussian mixtures in variational Bayesian approximation, *Journal of Machine Learning Research*, 7:625-644, 2006

[87] M.S. Waterman and M. Vingron, Sequence comparison significance and Poisson approximation, *Statistical Science*, 9(3):367-381, 1994

[88] C.L. Wei, A global map of p53 transcription factor binding sites in the human genome, *Cell*, 124(1):207-219, 2006

[89] G. Wahba, *Spline models for observational data.* Philadelphia: Society for Industrial and Applied Mathematics, 1990

[90] S.G. Walker, On sufficient conditions for Bayesian consistency. *Biometrika* 90, 482-488, 2003

[91] S.G. Walker, New approaches to Bayesian consistency. *Ann. Statist.* 32, 2028-2043, 2004

[92] S.G. Walker, Bayesian consistency for a class of regression problems. *South African Statistical Journal* 37, 149-167, 2003

[93] S.G. Walker, A. Lijoi and I. Prunster, On rates of convergence for posterior distributions in infinite-dimensional models. *Ann. Statist.* to appear, 2007

[94] N.A. Woods and N.P. Galatsanos, Non-stationary approximate Bayesian super-resolution using a hierarchical prior model, *IEEE International Conference on Image Processing,* 1, 37-40, 2005

[95] C.F.J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95-103, 1983

[96] H. Xue and V. Govindaraju, Hidden Markov models combining discrete symbols and continuous attributes in handwriting recognition, *IEEE-PAMI*, 28(3):458-462, 2006

[97] S. Yu and H. Kobayashi, An efficient forward-backward algorithm for an explicit-duration hidden Markov model, *IEEE Signal Processing Letters*, 10(1):11-14, 2003