Bayesian Inference and High-D space Characterization with application in Paleoclimatology and Biology

by

Luan Lin

B.S., University of Science and Technology of China, Hefei, China, 2006

M.S., Georgia Institute of Technology, Atlanta, GA, 2008

Sc.M, Brown University, Providence, RI, 2009

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

Providence, Rhode Island

May 2012

Abstract of "Bayesian Inference and High-D space Characterization with application in Paleoclimatology and Biology" by Luan Lin, Ph.D., Brown University, May 2012

This thesis is a mathematical study of paleoclimatology and computational biology.

Part I gives an introduction and overview of this dissertation.

Part II presents the applications of hidden Markov model in paleoclimatology study.

Chapter 1 employs a two state hidden Markov model with multivariate emission to challenge the assumption that ENSO-like variability dominated Peru margin oceanography on decadal and longer time scales over the Holocene epoch. The HMM result shows that two regimes of variability dominate, one shows strong correlations between surface and subsurface proxies while the other does not.

Chapter 2 is another application of hidden Markov model in paleoclimatology. A pair hidden Markov model is built to implement the alignments between pairs of stratigraphic records and provide uncertainty analysis. In addition to the most probable alignment, centroid alignment is also investigated.

Part III focuses on characterization of high dimensional space.

Chapter 3 develops an iterative algorithm to characterize the high dimensional spaces by identifying the most informative variables through mutual information based criteria. Application to the posterior space of RNA secondary structure is presented.

Chapter 4 proposes a novel model based clustering algorithm based on L_p distance in the situations where not only samples but also corresponding densities/probabilities (or subject to a normalization constant) are available. We demonstrate the abilities of this approach by experimenting via Gaussian Mixture Models.

Part IV makes some conclusions and suggests future directions.

This dissertation by Luan Lin is accepted in its present form
by the Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____
Charles Lawrence, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____
William Thompson, Ph.D., Reader

Date_____ _____
Timothy Herbert, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____
Peter M. Weber, Dean of the Graduate School

The Vita of Luan Lin

Luan Lin was born in Fuzhou, Fujian province, China, on June 28, 1984.

She graduated from the Middle Schoold attached to Fujian Normal Univerisy. In 2002, she started her undergraduate study in University of Science and Technology of China and received her Bachelor of Science degree in Mathematics four yesars later.

She came to the United States in 2006 and studied in the School of Mathematics at Georgia Institute of Technology. Two years later, she received her Master of Science degree in Mathematics and then started her Ph.D. program in the Division of Applied Mathematics at Brown University. During the Ph.D. program, she received a Master of Science in Applied Mathematics in 2009.

This dissertation was defended on April 20th, 2012.

Acknowledgments

I would like to thank all of the people who have helped me to complete my thesis.

Most of all, I want to extend my deepest gratitude to my advisor, Prof. Charles Lawrence, for his invaluable guidance, support, and encouragement throughout my graduate studies. His inspiring ideas and valuable suggestions shaped my thought and guided my career. This thesis would not have been possible without his persistent help.

I would also like to thank my other committee members, Prof. William Thompson and Prof. Timothy Herbert, for reading my dissertation and for giving me valuable comments.

Next I would like to thank all the faculty members in the Division of Applied Mathematics. I owe special thanks to Prof. Stuart Geman and Prof. Matthew Harrison, who shared their valuable opinions and have been very helpful during my research. I want to thank the members of my Lab group whom I worked with so closely during my time at Brown, Luis E. Carvalho, Eric Ruggieri, Jessica Nadel, Lauren Alpert and Matthew Parks. Also, I want to thank Yi Cai, Xinghui Zhong and all of my friends in the Division of Applied Mathematics for helping me in every aspect.

I also want to take this opportunity to thank Prof. Yang Wang in Department of Mathematics at Michigan State University and Prof. Hao Min Zhou in School of Mathematics at Georgia Institute of Technology. The one year research experience with them are quite valuable and I thank them for their help and understanding. I also owe my thanks to my friends in Georgia Institute of Technology, who offered a lot of help during my first two years in the United States. Specially, I thank Hao Deng, Hua Xu, Jing Xu, Ruoting Gong, Yun Gong and Huy Huynh.

Last but certainly not least, I owe my thanks to my family. Thanks to my parents Daiwei Lin, Fen Chen and my sister Nan Lin, who support me spiritually. I also thank Lo-Bin Chang for his love, companion and encouragement. Without their everlasting understanding, this endeavor would have been much more challenging.

Contents

List of Figures

List of Tables

Part I

Introduction and Overview

Increasingly, scientists in many fields including finance, biology and geology are concerned with the task and challenge of modeling and analyzing high-volume and high-dimensional data. Clustering is usually the first step to characterize a high dimensional space. In addition, a lot of statistical models have been developed to model complex process and data, among which the hidden Markov model (HMM) is widely applied.

In this thesis I present two applications and a theoretical development of methods for drawing inferences in discrete high-D spaces. Probabilistic models used in these spaces often can return not only samples from the posterior space, but also the probability of the sampled value or at least a variable proportional to the sampled value. In Chapter 4 I describe a novel clustering procedure that uses both the sampled values and their probabilities. As my collaborators and I are in the process of preparing papers associated with each of the four components of this research, I use material from the draft of these papers in what is presented below.

Application of hidden Markov model

Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging and biological sequence analysis. The basic theory of HMMs was first described in a series of classic papers by Baum and his colleagues [10, 11, 12] in the late 1960s and early 1970s. Over 50 textbooks on HMMs are now available including ones concerning topics such as finance, hand writing, traffic prediction, proteomics, data mining, and animal vocalization. A large number of papers on HMMs are contributed to the speech recognition literature, where they were applied first in the early 1970s. One of the best general introductions to the subject is the tutorial in Rabiners paper [5]. In the first two chapters of this thesis I describe the application of HMM to stratigraphic data sequences of proxies for historic climate variables.

HMMs are so far the most appropriate and well studied technology for the analysis of sequences in which one believes that substrings of observations within the long

input sequence belong to a finite number of states of unknown length and location. In a two state HMM, the goal is to assign each data point in the input sequence to one of the two states using only the input data sequence. The key difficulties in meeting this goal are that there are no direct data on the state at each data point, and for all but the shortest sequences there are a very large number of ways to assign data points to the states. For example, with a two state HMM applied to a sequence of N data points there are 2^N possible assignments.

To address these challenges an HMM employs two sets of variables: the variables describing observed data, here the proxy data, at each data point called "emission variables"; and the state variables that indicate the state assigned to each data point. The state variables are called "hidden variables" because there are no direct observations of these variables. The HMM model assumes that the underlying state sequence is a Markov chain and all the information contained in the observations should be conveyed by this Markov process. HMM is one example of graphical models, in which two constraints are imposed. The first constraint requires that the underlying state sequence $\{X_n\}$ is a Markov chain, and the second constraint stipulates that for the observation sequence $\{Y_n\}$, Y_n is independent of other variables given X_n. We have the following model:

$$P(\{X_n, Y_n\}_{n=1}^N) = P(X_1) \prod_{n=2}^{N} P(X_n|X_{n-1}) \prod_{n=1}^{N} P(Y_n|X_n),$$

where $\{X_n\}$ is the underlying state sequence and $\{Y_n\}$ is the observation sequence. Although there exists the constraint that the underlying process is a Markov chain, it has been shown that the HMM is dense in the space of finite-state discrete-time stationary processes [51]. Hence, the HMM is universal and robust, which might explain the success of application in many fields.

In this thesis, we are concerned with the HMM with discrete-time and continuous observations. There are two sets of parameters for such an HMM: the parameters of the transition matrix $A = (A_{ij})$, initial distribution π, and the parameters for the likelihood model in $P(Y|X)$. The HMM is well developed and there are many

extensions, including pair HMM, profile HMM and factorial HMM etc. HMM could also be generalized to allow continuous state spaces. Generalized hidden Markov model (GHMM), a particular extension of HMM will be of particular interest to us. In GHMM, multiple observations are made at each time step instead of one, and the number of observations emitted could be random and modeled by discrete distribution like geometric distribution.

HMM is well developed, and learning and inference of HMM are well studied. If the training data is available, maximum likelihood estimate (MLE) could be employed. In the situations with prior knowledge, i.e. prior distribution, maximum a posterior estimate (MAP) can be applied. Baum-Welch algorithm, a particular form of the well known expectation maximization (EM) algorithm is widely used for parameter learning, and does not require a set of training data in which not only the emission but also the hidden states are known. With minor modification in forward algorithm, viterbi algorithm has been developed to find a single most probable state path globally.

As the first part of this thesis, two applications of hidden Markov model to paleoclimatology are presented.

In Chapter 1, a two state hidden Markov model with multivariate emission is employed to challenge the assumption that ENSO-like variability dominated Peru margin oceanography on decadal and longer time scales over the Holocene epoch (last 11ka). Interestingly, with my collaborators I find strong evidence for two regimes in the coupling between atmosphere and the ocean, with one showing substantial correlations between proxy variables and the other showing very little correlation. Our findings that the switching between these states, is rapid, frequent, and well defined, suggest that the future of the earth's climate over the next decades may be much less predictable than previous thought.

In Chapter 2, a pair hidden Markov model is developed to align pairs of climate response observed in different marine cores. In our pair hidden Markov model, the hidden states are different sedimentation rates, emissions are the sum of squared proxy residuals after aligning the linear interpolated sequences. Here, transition

probability reflects the preference of smooth transitions between different sedimentation rates. With this HMM we employ stochastic back sampling to draw samples from the posterior distribution of the states, which we use to put Bayesian confidence limits on the alignments at each point in the stratigraphic record. Interestingly, these results show that the MAP (or MLE), Viterbi alignment, often falls outside the confidence limits. We also find that the centroid estimator which returns the median alignments, does not exhibit this untoward behavior.

Characterization of High-D space

Curse of dimensionality has been known for decades and its impact has been felt in many different fields. Multiple dimensions are hard to enumerate especially when the dimension is really high, and are difficult to visualize. In statistics, there are a lot of problems with parameter estimation and density approximation due to the paucity of data. Computation complexity is another issue involved in high-dimensional inference because of the exponential growth of the number of possible values with each dimension. For a discrete N-dimensional space where N is large, there are 2^N mass points even when the cardinality for each dimension is only two. Therefore the exact calculation on this space is often NP complete, e.g. expectation, variance, maxima point, etc.

The posterior space of discrete high-dimensional space often shows multimodality. For example, Ding et al. [29] has shown that the energy landscape of RNA secondary structures is often complicated, and the ensemble in the Boltzmann-weighted model usually contains multiple well separated structural clusters. Other evidence of multimodal exists in image computing, handwriting recognition and topics in a document etc. This leads to the important question of how to characterize such complex posterior spaces.

Clustering is the assignment of a set of data into several subsets so that data in the same cluster are similar to each other in some sense. Clustering is a method of unsupervised learning and is commonly used for statistical data analysis in dif-

ferent fields. There have been a lot of well built algorithms on clustering problems. Hierarchical algorithms iteratively find clusters using the clusters established in the previous step. Partitioning algorithms construct various partitions and then evaluate them by some criteria. Density-based clustering algorithms are devised to discover arbitrary-shaped clusters, in which a cluster is regarded as a region in which the density of data objects exceeds a threshold. Subspace clustering methods look for clusters that could only be seen in some subspaces, i.e. in a particular projection of the data. The problem of subspace clustering is given by the fact that there are 2^d different subspaces of a space with d dimensions. There are also statistical model based clusterings, e.g. Mixture Model clustering. Probabilistic finite mixture modeling [63] [34] is one of the most popular parametric clustering methods. Several probabilistic models like Latent Dirichlet Allocation [13] and Gaussian Mixture Model (GMM) [34] have been shown to be powerful in various applications. However, parametric mixture models are effective only when the underlying distribution of the data is either known, or can be accurately approximated by the distribution assumed by the component model.

In Chapter 3, we construct a characterization criteria based on entropy measure and design an iterative algorithm to identify the most informative variables in order to describe the shape of the high dimensional space. We apply this method to the posterior space of RNA secondary structure to identify the crucial base pairing during the folding process and provide an approach to cluster the posterior space. A comparison with Sfold, a nucleic acid folding software package, is provided.

When using probabilistic models we can often not only draw samples of the unknowns, but also evaluate the probabilities or densities of samples, $p(x)$'s (or evaluate them up to a normalization constant). Unlike traditional clustering algorithms in which the $p(x)$'s are unused, in Chapter 4 we propose a novel model based clustering algorithm based on empirical L_p distance in which we use both x's and $p(x)$'s. We argue that this algorithm is more robust and can handle various applications via adjusting the power p of L_p distance. Moreover, we provide an approach to figure out the number of underlying substantial clusters and a way to diminish local minimiza-

tion problem. In addition, this method is very general and applicable to numerous parametric models.

Part II

Application of hidden Markov model (HMM) in Paleoclimatology Study

Chapter 1

Flickering switch of Holocene oceanography along the Peru margin

The 4-7 year El-Nino-Southern Oscillation (ENSO) drives interannual climatic variability across the tropical Pacific and spawns anomalies in sea surface temperature (SST) and precipitation elsewhere. Climatologists and paleoclimatologists frequently assume that ENSO variability can be scaled up to describe the state of the tropical Pacific on timescales from decades to millions of years. We report evidence from very high-resolution multidimensional paleoclimatic time series that challenges the assumption that ENSO-like variability dominated Peru margin oceanography on decadal and longer time scales over the Holocene epoch (last ∼11 ka). We find that two Holocene regimes of variability dominate; one shows convincing correlations consistent with the coupling between surface and subsurface proxies while the other does not. ENSO variability operates in a regional, top-down fashion, by varying exchanges of heat, momentum, and moisture between the overlying atmosphere and the sea surface of the tropical Pacific. In contrast, most of the correlations we find between SST, biological productivity and subsurface temperature and ventilation, for either Holocene regime, resemble an ENSO-like climate pattern. Instead, they likely originate from outside the tropics through remote, bottom-up forcing. Since

phase transitions in the late Holocene are rapid (< 20 yrs), with average length of 190 yrs and other evidence indicates the earth has be in the high correlation regime since 1800, the present system could undergo sudden shift at any time.

1.1 Motivation

The robust evidence that paleoclimatic variability in the eastern equatorial Pacific (EEP) increased in the latter half of the Holocene is nearly uniformly interpreted as a sign that ENSO variability itself increased at about 5,000 ybp [72, 64, 50, 22, 58]. Indeed, it has been proposed ENSO-like patterns can arise on decadal to orbital (10^4) year time scales by asymmetries in the amplitude of extreme El Nino or La Nina events and through changing biases that make the EEP more disposed to either El Nino or La Nina regimes over time [79, 20]. Such changes could be driven during the Holocene through evolution of orbital forcing over the EEP, which by varying the seasonal distribution of insolation is predicted to alter the position of the intertropical convergence zone over the course of millennia [20, 37].

It is important to note, however, that the paleoclimatic time series taken as evidence for changes in the ENSO regime of the eastern equatorial Pacific typically record only one of the ensemble of variables coupled in the ENSO system. For example, a marvelous record for past changes in large-scale runoff into an Ecuadorian lake does capture decadal and longer variability in extreme rainfall in a region sensitive to El Nino related precipitation events [64]. However, such records do not address the occurrence at these time scales of regionally coupled changes in sea surface temperature, atmospheric convective activity, and oceanic upwelling expected from an ENSO mode of behavior. Examining the oxygen isotopic composition of individual planktonic foraminifera in the Galapagos, Koutavas and colleagues [49] inferred that SST variance, interpreted as increased ENSO variability, increased dramatically in the mid-late Holocene. Without other proxy data that resolve E-W spatial gradients in SST, and variations in the depth of the thermocline, the evidence for increased late Holocene SST variance in the Pacific "cold tongue" does not, however, unequivo-

cally favor ENSO as the underlying cause. Furthermore, as is typical for most marine sediments with modest deposition rates, samples average conditions over timescales much longer than the ENSO cycle.

Here we explicitly address the question of whether centennial- to millennial-scale oceanographic anomalies recorded in Holocene sediments along the Peru margin resemble the modern ENSO system. We focus on records from sediment core MW8708-PC2, retrieved from the central Peru margin (15.1°S, 75.7°W, water depth of 250m), since sediment at this site accumulated at a strikingly high and uniform rate (70cm/kyr) across most of the Holocene (10 - 1.4ka), and frequently contains annual laminations [19]. Sampling was designed to generate a multi-proxy suite of data with the potential to capture coupled ENSO dynamics if they were to persist into decadal and longer patterns during the Holocene (Figure 1.1). We obtained information on past SST through the alkenone proxy, on biological productivity through analyses of the abundance of alkenones (representing haptophyte algal productivity) and the percentage of organic nitrogen (a composite of all biological inputs to the sediment), and subsurface properties through δ^{15}N analyses (subsurface oxygenation/denitrification). High-resolution δ^{18}O analyses of benthic foraminifera (temperatures at 250m) were also taken, but foraminifera were not present in many intervals of the sediment and thus the δ^{18}O record is incomplete (see below). Samples of 0.5 cm thickness were obtained at 1.5 cm increments for alkenone analyses and at 2 cm increments for δ^{15}N and %N; the samples thus have an spacing of 20 to 30 years and average about 7 years of accumulation.

We included proxies for subsurface properties over the course of the Holocene, as these have received relatively little attention in prior work. Nitrogen isotopes provide a proxy for water column N-loss via dissimilatory nitrate reduction, which occurs only under conditions of extremely low subsurface oxygenation [21, 53, 54, 2]. N-loss can vary due either to regional changes in biological productivity and the rain of organic matter or to changes in the ventilation of the subsurface "shadow zone" [44] by changing the flux and/or properties of source waters [80, 23]. That positive excursions in δ^{15}N record the intensification of subsurface dysaerobic conditions is

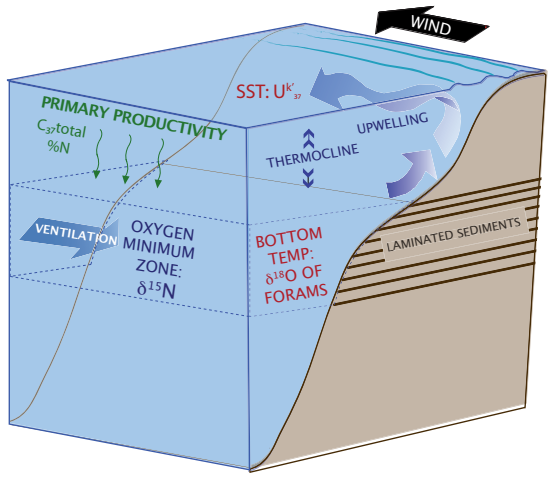


Figure 1.1: Idealized cross section along the Peru margin, showing major influences on surface and subsurface oceanography and biogeochemistry, and their expression in proxies captured in underlying sediments.

confirmed by shifts in the benthic foraminiferal populations in the core to nearly monospecific low-O_2 adapted assemblages in intervals of anomalously positive δ^{15}N (low inferred oxygen) (Chazen et al., in prep.).

From modern observations, the down core data should show the following associations between proxies if decadal and longer variability along the Peru margin followed the ENSO mode model. In the eastern tropical Pacific, El Nino climate phases see a warming of SST, a deepening of the mixed layer, a reduction in upwelling-favorable winds and concomitant crash in biological productivity [43]. In addition, a weakening of the oxygen minimum zone underlying the surface waters [16, 38], as a consequence of both a deepening of the oxygenated mixed layer and a decrease in the carbon flux driving oxygen consumption [7]. During periods of more frequent or protracted La Nina conditions, the proxy relations would reverse, coupling cool SST, with both higher productivity and greater intensity of the oxygen minimum zone. Although the mixed layer should deepen during El Nino events, it seems unlikely that temperatures at the 250 m bottom depth of our core site would be affected by anomalies

generated in the mixed layer.

A visual analysis of surface and subsurface proxies suggested that the correlations of the surface and subsurface proxies varied over time in an apparently abrupt manner (Figure 1.2). The second plot in Figure 1.2 gives the scaled proxy plots, in which for each proxy we subtract their mean and divide by their standard deviation. This leads us to consider a statistical approach suitable for capturing phase transitions in climate records. Specifically, we hypothesize that the variation among the four variables: SST, the two productivity proxies (C_{37}, %N), and subsurface oxygenation (δ^{15}N) follows two regimes, i.e. is biphasic, with phase transition times to be inferred from the data. We test this hypothesis using a Hidden Markov model (HMM). The first approach coming to mind might be the change point method or hidden Markov model (HMM), both exist to model non-stationary records. In our setting we favor hidden Markov model to change point method due to the following reasons. First, we are only interested in looking at high correlation areas and low correlation areas along the records; second, the number and frequency of change points are unknown. From this point of view, hidden Markov model is a better fit to model our problem. Therefore, we address this using a two state hidden Markov model, one for high correlation and the other for low correlation.

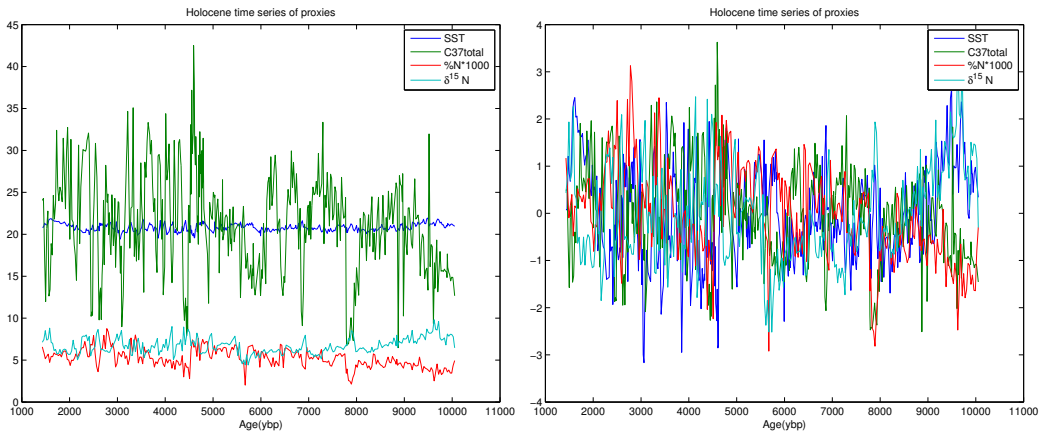


Figure 1.2: Holocene time series of proxies

1.2 Two states hidden Markov model

Given our collaborators' conjecture that there were two regimes, it is natural to start with a two state hidden Markov model, in which one state corresponds to the high correlation regime, the other for the low correlation regime. In this hidden Markov model, we examine the four observed variables, the proxies: alkenone-derived SST; C_{37} and %N productivity estimates; and the δ^{15}N subsurface dentrification proxy. We use one multivariate normal distribution to describe all the proxy reading assigned state one and another multivariate normal distribution to describe the variables in state two, as the "emission models" of the HMM. We examine QQ-norm plots of each of these variables to check our assumptions of normal distribution. As Figure 1.4 shows, we found little departure from the expected straight lines for normally distributed random variables of each of the proxies. Figure 1.3 gives the structure of this model. As indicated in Figure 1.3, both models are considered for every reading. Hidden variables, which are inferred by the HMM, of each reading indicate which of the two emission models is applicable at each reading, i.e. the state of the system at each reading.

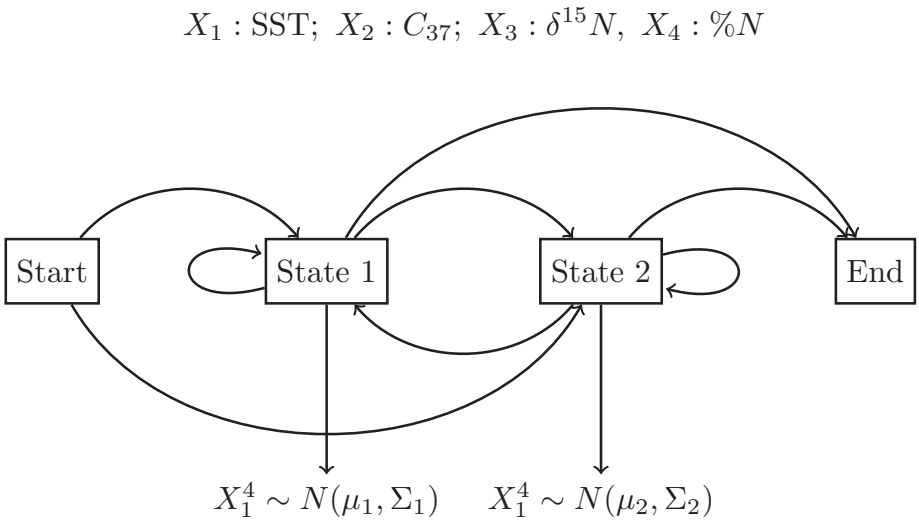


Figure 1.3: two state hidden Markov model with multivariate observation

In summary, this two state hidden Markov model can be described as follows. We

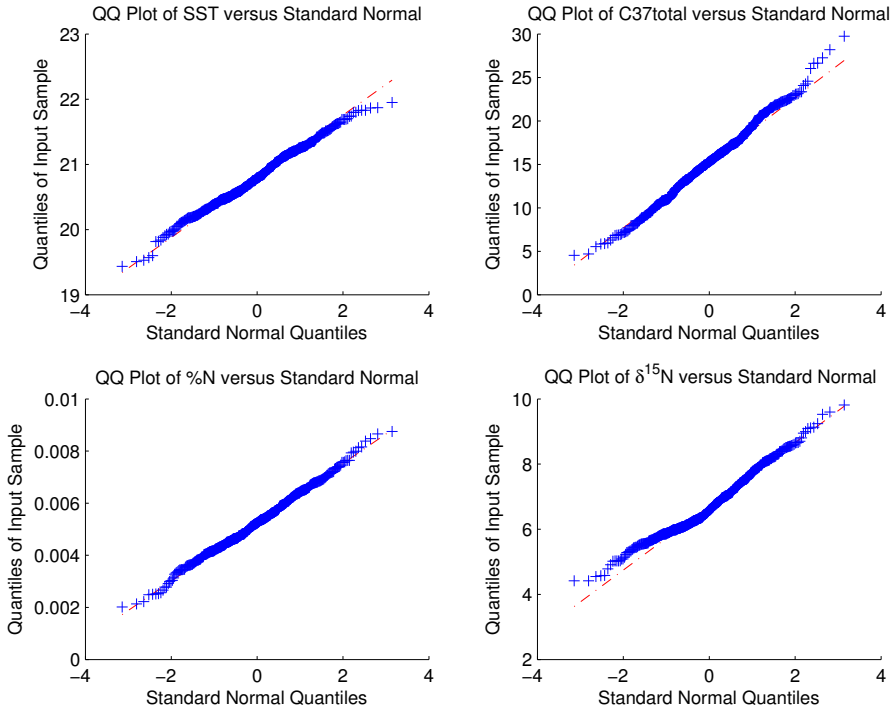


Figure 1.4: QQ-plot of the four proxies versus normal distribution

employ a multivariate normal distribution for these four variables in each of the two states as the emissions of this HMM, i.e. $Y_i|X_i \sim N(\mu_i, \Sigma_i)$. Here we use sequence $X = (X_1, X_2, \cdots, X_n)$ to denote the hidden state variable, and $Y = (Y_1, Y_2, \cdots, Y_n)$ to denote the observation sequence, i.e. the four proxies. The parameters of this model are means and variances of each proxy, and correlations among variables along with the transition probabilities between these two states for the underlying Markov chain. Let $a = (a_{ij})$ be the transition probabilities between state 1 and state 2, π be the initial probability for the underlying Markov process. Then if we know the state assignment of the underlying process, the likelihood of the sequence is

$$f(X) = \pi(X_1) \prod_{i=1}^{n} a_{X_{i-1}X_i}.$$

Then the joint probability of the observed sequence Y and a state sequence X given

the unknown parameters μ_i's and Σ's is:

$$f(X, Y|\theta) = a_{0X_1} \prod_{i=1}^{n} f_{X_i}(Y_i) a_{X_i X_{i+1}},$$

where $\theta = (\mu_1, \Sigma_1; \mu_2, \Sigma_2)$ and $f_{X_i}(Y_i)$ is the density for the multivariate distribution with parameter in state X_i.

When the paths are unknown, there is no longer a direct closed-form equation for the estimated parameter values, and some form of iterative procedure must be used. There is a particular iteration method that is standardly used, known as Baum-Welch algorithm [12]. The Baum-Welch algorithm calculates A_{kl} as the expected number of times each transition or emission is used, given the training sequences. To do this it uses the forward and backward values as the posterior probability decoding method. The probability that a_{kl} is used at position i in sequence x is

$$P(X_i = k, X_{i+1} = l|Y, \theta) = \frac{f(k, i) a_{kl} f_l(Y_{i+1}) b(l, i+1)}{P(Y)} \tag{1.1}$$

From (1.1) the expected number of times that a_{kl} is used is

$$A_{kl} = \frac{1}{P(Y)} \sum_i f(k, i) a_{kl} f_l(Y_{i+1}) b(l, i+1)$$

where $f(k, i) = P(Y_1, \cdots, Y_i, X_i = k)$ is the forward variable, i.e. the probability of the observed sequence up to and including Y_i, requiring that $X_i = k$; and $b(l, i) = P(Y_{i+1}, \cdots, Y_L|X_i = l)$ is the corresponding backward variable, i.e. the probability of the observed sequence after Y_i conditioning on $X_i = l$. $f(k, i)$ and $b(l, i)$ could be easily calculated by the well known forward and backward algorithm using the following recursion.

$$f(l, i+1) = f_l(Y_{i+1}) \sum_k f(k, i) a_{kl}$$

$$b(k, i) = \sum_l a_{kl} f_l(Y_{i+1}) b(l, i+1)$$

Having calculated these expectations, the new model parameters for transition prob-

abilities are calculated using

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}$$

For the multivariate parameters μ_i and Σ_i, we simply employ stochastic Expectation Maximization (EM). We first draw samples from the posteior space using backward sampling algorithm, and choose the parameters which maximize the total likelihood of these sample paths, i.e. the maximum likelihood estimate (MLE), or maximum a posterior estimate (MAP) if prior knowledge of the parameters is available and modeled as some prior distribution. MLE and MAP estimation possess a number of attractive asymptotic properties. As the sample size is large enough compared to the number of unknowns, sequences of MLEs have consistency (a subsequence of the sequences would converge in probability to the value being estimated, i.e. the true value), asymptotic normality and efficiency (no asymptotically unbiased estimator would have lower asymptotic mean squared error) properties. However, if the sample size is not large enough, then there is nothing to assure that MLE and MAP will have these favorable properties.

This hidden Markov model outputs the predicted regime at each point in the record, the times of transition between regimes, and indices of confidence of these two. Figure 1.5 gives the most probable state path and also a comparison of the most probable path and centroid path, Tables 1.1-1.3 gives the resulting parameter estimates for μ_i and Σ_i and Figure 1.6 gives the probability that the climate system is in regime 1 versus regime 2 for each data point. Centroid estimator is the nearest feasible solution to the center of mass of the posterior space and represents the central tendency of the space. In this example, the centroid path is very close to the most probable path. However, it is not always the case. In Chapter 2, we will discuss more on centroid estimator and will see some examples where the centroid is very different from MLE or MAP.

Table 1.10 gives the comparison of findings with ENSO dynamics. Upscaling ENSO couplings to decadal and longer time scale variability makes predictions on proxy correlations that can be tested via the HMM analysis. It predicts that SST

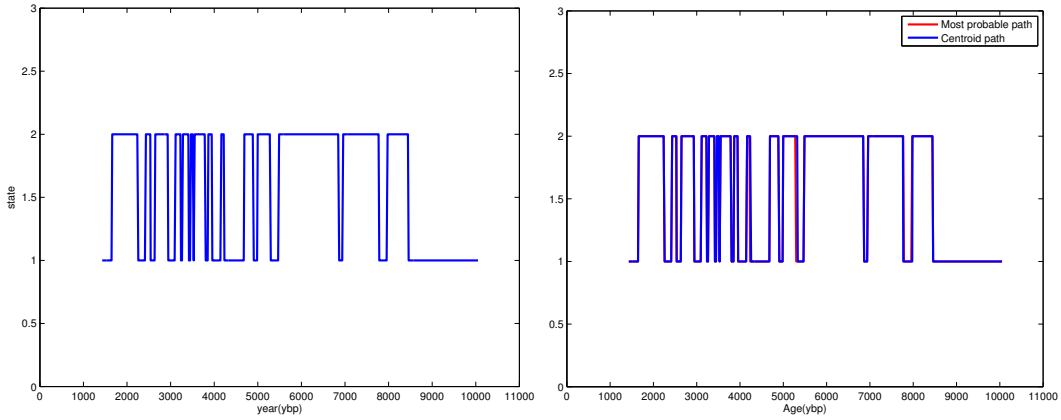


Figure 1.5: Most probable regime path returned by this hidden Markov model and compared with the Centroid regime path

Table 1.1: Results of Hidden Markov Model: Mean and Variance parameter for two states

	Mean				Variance			
	SST	C_{37}	δ^{15}N	%N	SST	C_{37}	δ^{15}N	%N
Regime 1	20.826	20.139	7.559	0.0047	0.2546	44.578	0.4449	1.1e-6
Regime 2	20.793	22.005	6.117	0.0057	0.1376	25.477	0.2552	9.6e-7

Table 1.2: Results of Hidden Markov Model: Covariance parameter for two states

	SST; C_{37}	SST;$\delta^{15}N$	SST;%N	C_{37}; $\delta^{15}N$	C_{37}; %N	$\delta^{15}N$; %N
Regime 1	-1.3595	0.0966	-0.0002	-1.3651	0.0040	-0.0002
Regime 2	0.0279	-0.0183	0.0001	0.1563	-0.0005	0.0001

Table 1.3: Results of Hidden Markov Model: Correlation parameter for two states

	SST; C_{37}	SST;$\delta^{15}N$	SST;%N	C_{37}; $\delta^{15}N$	C_{37}; %N	$\delta^{15}N$; %N
Regime 1	-0.4036	0.2871	-0.3306	-0.3065	0.5886	-0.3559
Regime 2	0.0149	-0.0975	0.3267	0.0613	-0.1067	0.1573

Table 1.4: Comparison of findings with ENSO dynamics extended to decadal-century scale

Correlations	ENSO	Low Correlation Regime	High Correlation Regime
SST; C_{37}	−	∼0 (0.015)	− (−0.40)
SST; %N	−	+ (0.32)	− (−0.33)
SST; δ^{15}N	−	∼0 (0.061)	+ (0.29)
SST; δ^{18}O	− (∼0)	∼ 0 (−0.098)	+ (0.28)
δ^{15}N; C_{37}	+	− (−0.11)	− (−0.31)
δ^{15}N; %N	+	+ (0.16)	− (−0.36)

should be negatively correlated with the productively proxies (C_{37} and %N), δ^{15}N, and δ^{18}O. Our results shows little correlation between any of the proxies in regime 2 except for a positive correlation between SST and the productively proxy %N. While in regime 1 SST and both productivity proxies are negatively correlated as predicted by ENSO model, the correlations with SST and both δ^{15}N and δ^{18}O are positively correlated in contradiction with ENSO observation. Thus in both regimes our results do not support extending the ENSO model to account for the coupling between SST and two of the three other oceanographic parameters.

From Table 1.1-1.3, the covariance and thus corrleation between variables in state 1 is apparently higher than those in state 2, although some of them even present different directions of relationship. Figure 1.6 gives the probability that the climate system is in regime 1 versus regime 2 for each data point. As Figure 1.6 indicates, the probability of being in regime 1 is almost always near either zero or one at nearly all of the data points indicating that there is little ambiguity whether the system is in regime 1 or regime 2.

Probably the most difficult problem faced when using HMMs is that of specifying the model in the first place. There are two parts to this: the design of the structure, i.e. what states there are and how they are connected, and the assignment of parameter values. A likelihood ratio test clearly favors rejection of the null hypothesis that the data are described by a single multivariate (one regime) normal model (with p-value less than 10^{-6}) as opposed to this two regime HMM model, but does not favor rejection of two regimes in favor of three (with p-value = 0.062>0.05).

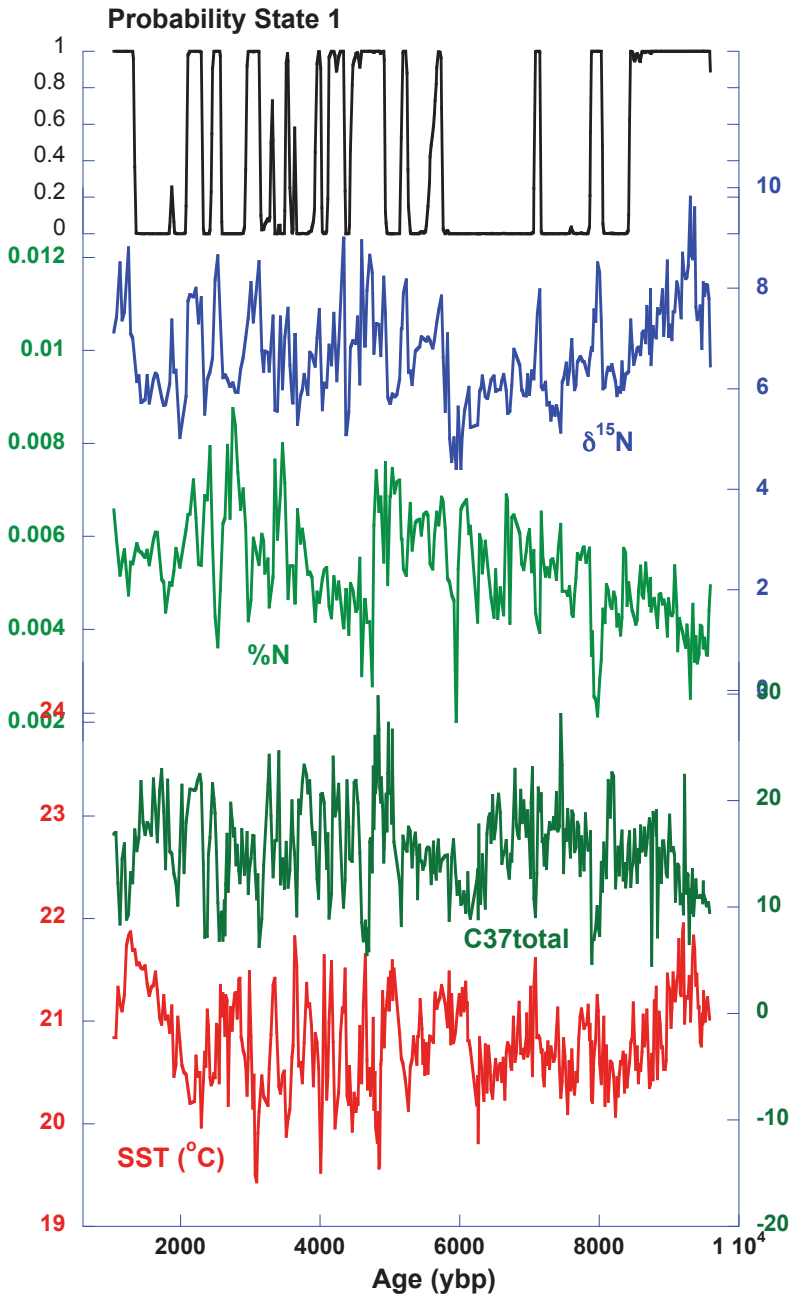


Figure 1.6: Holocene time series of proxies used in Hidden Markov analysis, along with probability (p) that the system lay in the correlated regime 1. The probability of regime 2 is 1-p.

Given the robust prior evidence of a change in the system at ∼5.5 ka, we also tested the hypothesis that two separate HMM (two regime) models for the intervals, one before and one after 5.5 ka fit the data better than single two regime HMM model for the full 11 kyr. The likelihood ratio test again strongly favored rejection of the simpler model with p-value less than 10^{-6}. Thus, these results support the hypothesis of a change in behavior at 5.5 ka. Furthermore, the HMM model that distinguishes between late and early Holocene behavior indicates that the variance in SST of both regimes doubles in the late Holocene (Regime 1: from 0.11 to 0.22 °C and Regime 2: from 0.069 to 0.1393 °C). Section 1.4 would report the result of this two HMMs model.

1.3 Interpretation of HMM results

Regime 1 represents a pattern of behavior in which SST, productivity and sub-surface oxygenation are either positively or negatively coupled; it also contains about twice the variance for these parameters as regime 2 (Table 1.1). Mean δ^{15}N, indicating a stronger oxygen minimum zone, is also higher in regime 1, whereas mean values for SST and productivity are not distinguishable between the regimes. Regime 2 presents no clear correlation between the proxies, but accounts for approximately 60% of the system behavior in the early Holocene, and 45% of the time after 5.0 ka. In the late Holocene, transitions between regimes occurred frequently (18 transitions in 4.5 ka), were rapid (< 20 years), well defined, short (mean length of 190 yrs), and of irregular duration (std of 140 years). These findings suggest that in the more recent past the southern tropical Pacific has frequently been subjected to abrupt and unpredictable phase transitions.

Table 1.5: Results of Hidden Markov Model: Parameter estimates for δ^{18}O

	δ^{18}O	δ^{18}O	δ^{18}O Correlation			
	Mean	Variance	δ^{18}O; SST	δ^{18}O; C_{37}	δ^{18}O; δ^{15}N	δ^{18}O;%N
Regime 1	1.6842	0.0245	0.2843	-0.4132	0.4413	-0.3517
Regime 2	1.6230	0.0095	0.0155	-0.0729	0.0652	-0.0976

δ^{18}O measurements from benthic foraminifera (Bolivina seminuda) provide for the first time strong evidence that major changes in subsurface density structure accompanied changes in oxygen minimum zone intensity during regime 1 (Figures 1.7). Benthic foraminifera are absent in many intervals, most commonly in zones with regime 2 behavior. Thus, δ^{18}O measurements were too episodic to be meaningfully incorporated into the HMM. However, we estimated the means δ^{18}O , its variances and its correlations with the other four variables for each regime within intervals in which there were benthic foraminifera, and present them in Table 1.4. The isotopic measurement incorporates the effects of both temperature and salinity. However, because lateral and vertical salinity variations in the depth range of the PC2 core location are modest [52], the δ^{18}O variations almost certainly result from temperature changes. The amplitude of the subsurface temperature changes (on the order of 1 - 3°C) is as large as the accompanying SST changes revealed by alkenone paleothermometry. The substantial positive correlation in regime 1 between δ^{18}O and δ^{15}N , in Table 1.5, indicates that subsurface temperatures shift to colder values (more positive δ^{18}O) at the same time that the oxygen minimum zone intensified (more positive (δ^{15}N) in this regime, while these two are nearly uncoupled in regime 2. Table 1.5 also shows that colder subsurface temperatures (positive δ^{18}O) couple to warmer SST in regime 1, while surface and subsurface temperatures are uncoupled in regime 2. The data thus record modulations in the surface to intermediate water temperature (and density) gradients in the coupled mode (regime 1), with the strongest stratification favoring the most intense oxygen depletion, as recorded by the most positive values of δ^{15}N and the presence of laminations in the sediments. These changes in both the temperature and chemistry of subsurface waters occur with no resolvable offset in time.

What is surprising is that correlations between surface and subsurface variables in Regime 1 do not clearly resemble the El Nino-La Nina oscillation. While the association of colder SST and higher productivity follows an upwelling-driven signal is consistent with ENSO dynamics, (Figure 1.1), the negative correlation of both productivity proxies with oxygen depletion (δ^{15}N) rules out the regional, top-down

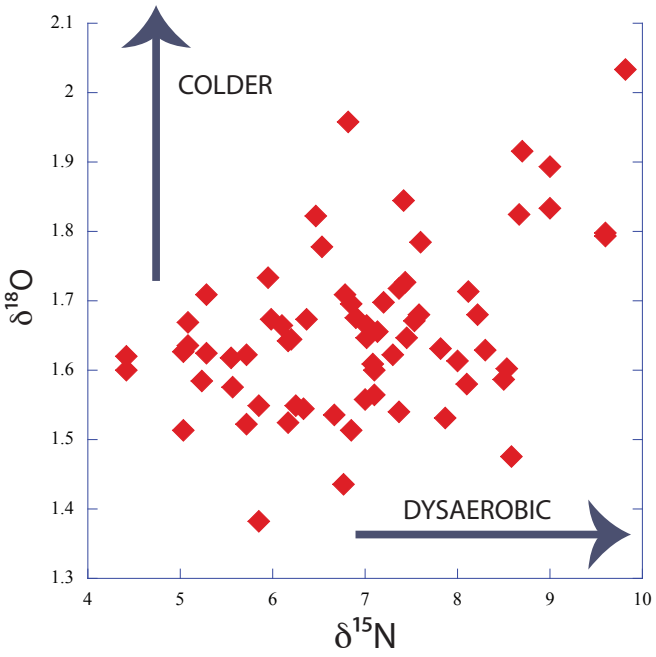


Figure 1.7: **Correlation between δ^{18}O of benthic foraminifera (Bolivina seminuda).** a proxy for temperature at the core depth of 250m, and δ^{15}N, a proxy for the intensity of denitrification in the oxygen minimum zone. The range of δ^{18}O values equates to about 3°C, if temperature is the only control on isotopic values. Note that the most dysaerobic conditions coincided with the incursion of colder waters at 250m depth.

control of the oxygenation and denitrification by sinking carbon flux observed on short time scales [54, 25]. The substantial positive correlation between δ^{18}O and δ^{15}N, and the negative correlation between productivity proxies and N-loss (positive δ^{15}N) in regime 1 indicates that oxygen depletion is more common in this regime under colder subsurface temperatures (higher δ^{18}O), and lower carbon flux, which paradoxically would be expected to promote higher O_2 contents in the oxygen minimum zone. Furthermore, the negative correlation of SST with N-loss (positive δ^{15}N) (Table 1.1-1.3) and temperature at 250m depth (positive with δ^{18}O) is not consistent with a downward propagation of mixed-layer anomalies, as occurs under modern ENSO conditions [16]. The paradox is resolved if we infer that changes in intermediate water ventilation rate, reflected by changes in sub-surface temperature, played the dominant role in controlling suboxic conditions along the Peru margin. Much poorer ventilation of source waters within the oxygen minimum zone (high δ^{15}N) apparently outweighed the reduced carbon flux in the low-oxygen end member, confirming the importance of physical oxygen renewal in regulating the OMZ on decadal-millennial timescales [47]. These subsurface temperature changes would have also affected the surface by changing density gradients and nutrient fluxes in the photic zone, explaining the regime 1 associations in (Table 1.1-1.3). We propose that Regime 1 therefore represents, bottom-up forcing of oceanographic and biological variability on decadal to centennial timescales along the Peru margin.

The link of regime 1 to perturbations in subsurface temperature and oxygen (denitrification) hints at a remote origin, whether at the southern edge of the subtropical gyre [35, 74, 46, 70] or perhaps poleward in the region forming Sub-Antarctic Mode Water that flows circuitously into the equatorial Pacific system of undercurrents [80, 66, 70].The OMZ of the eastern equatorial Pacific develops in the shadow zone below the strong eastward shoaling of the thermocline. Ventilated thermocline theory [44, 41] takes the depth of the mixed layer at the eastern boundary, and its slope into the subtropical gyre, as boundary conditions to derive the wind-driven circulation within the vast interior of the subtropical basin. The δ^{15}N and benthic temperature data suggest that this basin-wide stratification, at least at its eastern

boundary, has been perturbed strongly from below at centennial times scales over the last 11,000 years. At its extreme, shoaling of colder isotherms on the order of 100m coincided with reduced ventilation of the shadow zone, leading to the high δ^{15}N values typical of the Regime 1 intervals. One might infer that these perturbations, which may have counterparts in centennial to millennial paleoclimatic variability along the Antarctic Peninsula [77] and the southeast Pacific [33] affected the entire South Pacific subtropical gyre circulation.

Correlations between proxies in regime 2 are also inconsistent with ENSO-like dynamics at the decadal to centennial scale. One could imagine that ENSO variability would not strongly couple surface and subsurface proxies, but the poor association of SST with the C_{37} productivity proxy and positive correlation of SST with %N in the regime 2 (Table 1.1-1.3) shows little similarity to the modern observations of the biological response to ENSO variability at the Peru margin [7]. Since our sampling does not resolve variations at the 4-7 year ENSO timescale, it leaves open the relationship between the longer-term regime changes we have detected and the behavior of the ENSO system itself over the Holocene.

A change in sedimentation at 1800 A.D. in the central Peru margin includes a rapid switch to higher δ^{15}N values and abundant benthic foraminifera, suggesting that the modern system experienced a switch to Regime 1 behavior at that time. Our finding of rapid phase transitions between regimes with an average duration of 190 years in the late Holocene suggests that a change back to regime 2 could occur at any time. If properly tuned state-of-the-art coupled climate models can generate the biphasic behavior of intermediate ocean circulation detected in our proxy time, then it would be interesting to learn what series of processes determine the duration of regimes and the speed of regime switching. Our results also suggest that decadal to century climatic variability along the Peru margin cannot be modeled by scaling up ENSO dynamics, but rather may originate from different, and longer-lasting, ocean-atmosphere instability, or from unknown external forcing.

1.4 two HMMs for the early and late Holocene

Due to the robust prior evidence of a change in the system at 5.5 ka and distinguishable pattern difference that regime shifts became much more frequent in the late Holocene, in this section we are going to work on a two HMM mode, in which separate HMM is modeled for early and late Holocene. On the other hand, the likelihood ratio test strongly favor the rejection of the null hypothesis that these proxies is modeled by only one hidden Markov model with p-value 7.8e-12.

The algorithms and parameter estimation are exactly the same. Therefore we just present the result.

Table 1.6-1.7 give the parameter estimation of the hidden Markov model for late Holocene (after 5.5 ka) and Table 1.8-1.9 give the parameter estimation for early Holocene (before 5.5 ka). We see a big difference between these parameters and those from previous single hidden Markov model. In this two HMMs model, the overall variance and covariance in late Holocene are higher than those in early Holocene, although the correlation relationship isn't always the case. Also, even in each separate HMM, the correlation of state 1 is not always higher than state 2. Furthermore, same as previous model, the correlations between surface and subsurface variables in both state 1 and state 2 are not consistent with ENSO dynamics.

Table 1.6: Results of Hidden Markov Model for late Holocene (after 5.5 ka): Mean and Variance parameter for two states.

	Mean				Variance			
	SST	C_{37}	δ^{15}N	%N	SST	C_{37}	δ^{15}N	%N
Regime 1	20.667	22.515	7.038	0.0056	0.2296	39.4702	0.6606	8.9e-7
Regime 2	21.182	23.586	6.041	0.0064	0.0707	32.4024	0.0461	1.1e-6

Table 1.7: Results of Hidden Markov Model for late Holocene (after 5.5 ka): Correlation parameter for two states

	SST; C_{37}	SST;$\delta^{15}N$	SST;%N	C_{37}; $\delta^{15}N$	C_{37}; %N	$\delta^{15}N$; %N
Regime 1	-0.3625	0.0321	-0.0300	-0.2333	0.2085	-0.2843
Regime 2	0.2897	-0.2770	-0.2220	-0.3836	-0.3344	0.2383

Table 1.8: Results of Hidden Markov Model for early Holocene (before 5.5 ka): Mean and Variance parameter for two states.

	Mean				Variance			
	SST	C_{37}	δ^{15}N	%N	SST	C_{37}	δ^{15}N	%N
Regime 1	20.801	19.864	6.770	0.0047	0.1358	26.7642	0.8585	7.8e-7
Regime 2	21.109	16.820	5.120	0.0054	0.0382	7.8302	0.225	1.9e-6

Table 1.9: Results of Hidden Markov Model for early Holocene (before 5.5 ka): Correlation parameter for two states

	SST; C_{37}	SST;$\delta^{15}N$	SST;%N	C_{37}; $\delta^{15}N$	C_{37}; %N	$\delta^{15}N$; %N
Regime 1	-0.2318	0.6362	-0.2874	-0.3460	0.4093	-0.6793
Regime 2	-0.6799	0.3493	0.4133	-0.3647	-0.4295	0.54243

Table 1.10: Comparison of findings with ENSO dynamics extended to decadal-century scale – late Holocene

Correlations	ENSO	Regime 1	Regime 2
SST; C_{37}	−	−(−0.3625)	+ (0.2897)
SST; %N	−	∼0 (−0.0300)	− (−0.2220)
SST; δ^{15}N	−	∼0 (0.0321)	− (0.2770)
δ^{15}N; C_{37}	+	− (−0.2333)	− (−0.3836)
δ^{15}N; %N	+	− (−0.2843)	+ (0.2383)

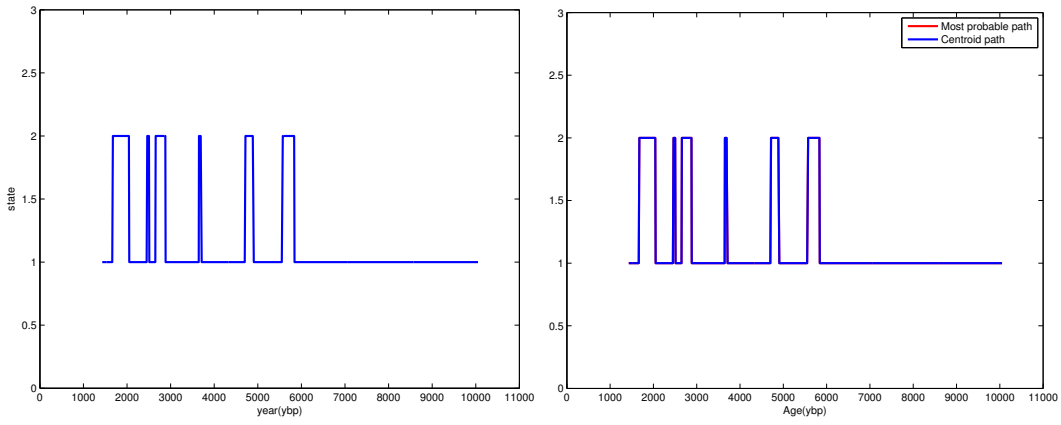


Figure 1.8: Most probable regime path returned by the two HMMs model and compared with the Centroid regime path

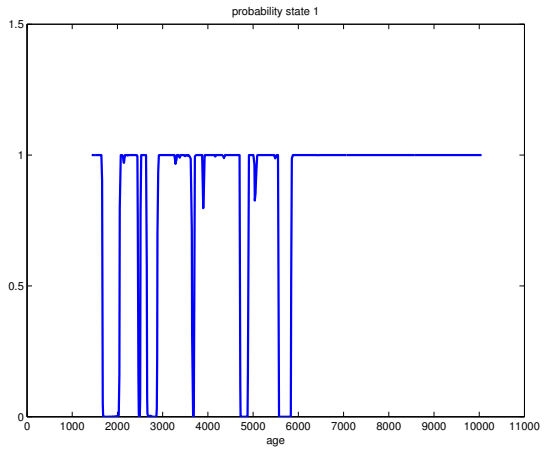


Figure 1.9: Probability of being in state 1 for the two HMMs model

Figure 1.8 gives the most probable state path returned by this two HMMs model and compare it with the centroid state path. Figure 1.9 plots the probability of each data point being in state 1. Compare with Figure 1.6, one similarity is that the regime shift is more frequent in late Holocene, although the assignment for the states at data points presents distinguishable difference.

In this section we only give a short summary of the result of this two HMMs model. More detailed analysis and interpretation of the result will be included in the next paper.

Chapter 2

Probabilistic Stratigraphic Alignment of Pleistocene δ^{18}O records

As pointed out by Lisiecki et al. in [55], paleoclimate studies are very crucial in analyzing climate change at current time in the context of natural climate changes in the past time. The common source of paleoclimate data is marine sediment core. Having a common time scale in which each record from various locations on the globe can be compared is essential to inference of the sequence on events. Fortunately, glaciation events have a global impacts. One of these alters the ratio of oxygen isotopes.

Oxygen is composed of 8 protons. Its most common form, which gives it an atomic weight of 16 (O^{16}) and is known as "light" oxygen, is composed of 8 neutrons. A small fraction of oxygen atoms, which is known as "heavy" oxygen, have two more neutrons and thus have an atomic weight of 18 (O^{18}). Ice in glaciers have less O^{18} than the ocean. However, the proportion of O^{18} also changes along with the change in temperature. The water-ice in glaciers came from the seawater as vapor originally. Later the vapor fell as snow and then became ice. During water evaporation, O^{18} is left behind and the water vapor is rich in O^{16}, since it is more difficult for O^{18}, the heavier molecule to evaporate. Therefore, the O^{16} is relatively enriched in glaciers,

while the oceans are relatively enhanced in O^{18}. The imbalance between these two isotope is more noticeable for colder climates than warmer climates.

The development of age models for paleoclimate records from marine sediment core is of central importance to establish the interactions between different sites of the climate system. The conventional way to build age models for long marine cores are either by aligning climate responses to orbital tuning, or aligning climate responses to other climate responses observed in other cores. So far, these tasks are mostly performed either manually, or by quantitative comparison [75, 69] and deterministic dynamic programming alignment algorithms [59, 55, 42]. A very important limitation of extant alignment methods is that they provide no information on the uncertainty of the alignments they produce. To address this shortcoming, in this chapter we develop probabilistic models and algorithms for the alignment of stratigraphic records.

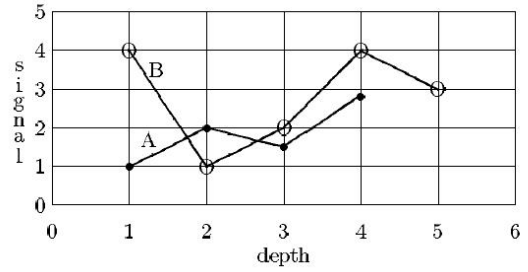
In addition to paleoclimate alignment, sequence alignment has been extensively employed in molecular biology and speech recognition to model the variations and similarities along sequences. Deterministic dynamic programming (DP) algorithms have been developed for alignments in all these three fields. In molecular biology, both global [65] and local alignments [78] were developed; in speech recognition, people employ dynamic time warp algorithms [81, 73]; and Lisiecki [55] developed stratigraphic alignment for paleoclimate records. Due to the fact that the deterministic dynamic algorithm only outputs a single optimal alignment without any uncertainty analysis, probabilistic models especially in the form of hidden Markov model have been extensively developed in other fields to address this limitation [4, 6, 45, 84, 31, 82, 8]. The purpose of the probabilistic model is to improve the performance of the alignments, and at the same time address underlying uncertainty and make statistical inference. Our first goal is to build probabilistic models for pairwise alignment of climate records based on the dynamic programming algorithm developed by Lisiecki & Lisiecki [55].

2.1 Match: Application of Dynamic Programming to the Correlation of Paleoclimate records

This section is an overview of Lisiecki's dynamic programming algorithm: Match [55].

Interpretation of diverse climate proxy records is very important in paleoclimate reconstruction. The correlation among stratigraphic records is often analyzed by matching corresponding signals. The dynamic programming algorithm implemented in "Match" is powerful in that it is able to compare all possible alignments between records and pick out the global optimal one. We illustrate signal matching by a simple example in which we would like to match series A to series B and the goal is to minimize the sum of the square of their difference in proxy values. Figure 2.1 is a good illustration of this procedure. In this example, we are aligning the 4 data points in series A to a subset of the 5 data points in series B. We built a table to record the square of difference between each point in A and each point in B and choose the alignment with the smallest sum based on this table.

a.



b.

series B \ series A	signal	1.0	2.0	1.5	2.8
	4.0	9.0	4.0	6.25	1.44
	1.0	0.0	1.0	0.25	3.24
	2.0	1.0	0.0	0.25	0.64
	4.0	9.0	4.0	6.25	1.44
	3.0	4.0	1.0	2.25	0.04

c.

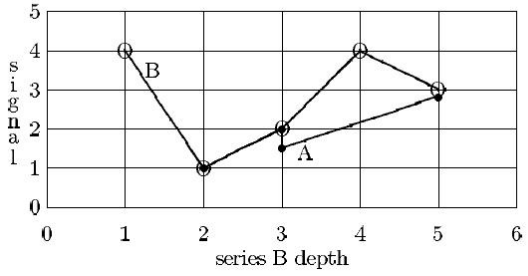


Figure 2.1: A simple Example. Figure cited from Lisiecki's paper [55].

The Match algorithm divides each series into several hundred sub-intervals and infers the alignment of these sub-intervals, each of which may contain one or more δ^{18}O data points. Match aligns these sub-intervals and in doing so allows points in one series to fall between points in the other series. Matching scores are calculated as the sum of squared difference of their corresponding, linearly interpolated value. Alignment is required because sedimentation rates vary between cores and change in time within cores. To address these differences in sedimentation rates, from one to five sub-intervals from sequence A is allowed to align with from one to five sub-intervals in sequence B in ratios of 1 to 5 up to a ratio of 5 to 1. The set of plausible mapping ratios includes: 1:4; 1:3; 2:5; 1:2; 3:5; 2:3; 3:4; 4:5; 1:1; 5:4; 4:3; 3:2; 5:3; 2:1; 5:2; 3:1; 4:1. The difference in ratios can be seen as compression and expansion of the two records with respect to one another. This procedure is illustrated as in Figure 2.2. As in the simple example above, the algorithm needs to build a table to record the scores of all possible alignments. Each table entry is filled recursively via dynamic programming using an algorithm much like those used to aligning biopolymer sequences, but with steps that concern expansion and contraction rather than insertion and deletion with the sum of all applicable penalties within the already matched intervals. After filling the whole table, the optimal alignment is selected to be the one with the lowest cumulative penalty using the back trace recursion.

There are a set of different penalties included to control differences and changes in expansion and contraction in the cumulative score, among which rate change penalty and rate penalty are most important. A rate change penalty is of the form $C_\delta n_r^2$, where n_r is the number of rate increments and C_δ is the rate change penalty coefficient. Here Match chooses a penalty proportional to n_r^2 in order to discourage fast changes between accumulation rates. Rate penalty is added when the accumulation rate is different from the "preferred" rate (a predefined preferred rate ratio, usually one to one). The rate penalty is of the form $C_\rho(\tilde{r}-1)^2$, where $\tilde{r} = \frac{1}{r}$ if $r < 1$ and $\tilde{r} = r$ if $r \geq 1$. No match penalty could also be added.

Now let us give the explicit recursion for this algorithm. Suppose there are m

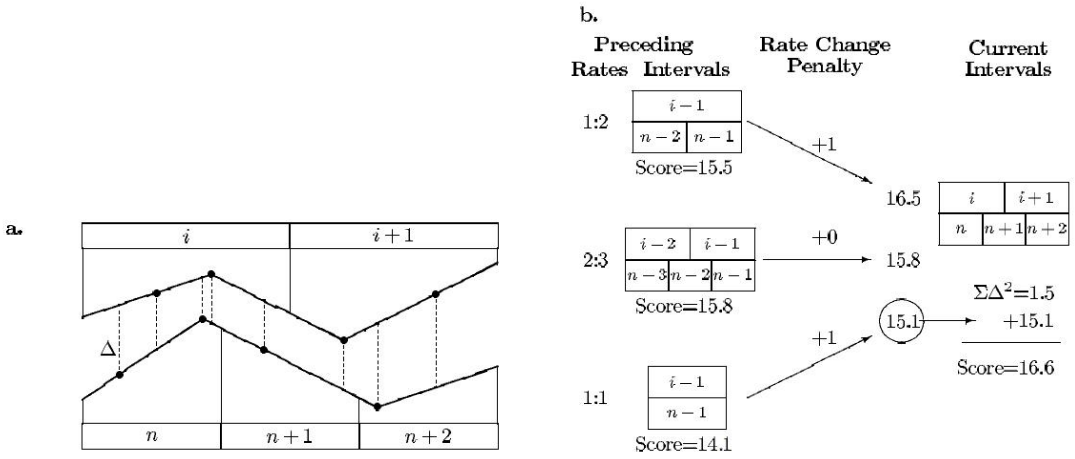


Figure 2.2: Matching two intervals in series A with three intervals in series B. Figure cited from Lisiecki's paper [55].

points in series A divided into I subintervals, and there are n points, J subintervals for series B. There are N_R number of possible ratios $R = \{r_1, r_2, \cdots, r_{N_R}\}$, which are used to align these two series. Let $S(i, j, r)$ be the cumulative penalty score of optimal alignment up to (i, j) (i^{th} interval in series A and j^{th} interval in series B) with ratio $r = a : b$. i.e. it maps intervals $i - a + 1, \cdots, i$ in series A to intervals $j-b+1, \cdots, j$ in series B. To initialize, for intervals with index less than the required number in corresponding mapping ratios, we set the penalty to be infinity in order to demonstrate the impossibility of such alignments. For intervals with index larger than or equal to the required number in corresponding mapping ratios, we add the sum of squared difference, rate penalty, and gap penalty together as their cumulative score.

a. Initialization: For every possible ratio $r = a : b$, set

   i. $S(i, j, r) = \infty$ for all $i < a$ or $j < b$

   ii. $S(i, b, r) = \rho(r) + \mu * (i - a) + \sum_{w \in \phi^r_{i,b}} \Delta^2_w$ for all $i \geq a$

   iii. $S(a, j, r) = \rho(r) + \mu * (j - b) + \sum_{w \in \phi^r_{a,j}} \Delta^2_w$ for all $j \geq b$

b. Recursion:

$$S(i,j,r) = \min_{r' \in R}[S(i-a, j-b, r') + \delta_{r,r'}] + \rho(r) + \sum_{w \in \phi^r_{i,j}} \Delta^2_w$$

where $\phi^r_{i,j}$ denotes the intervals $i-a+1, \cdots, a$ in A and intervals $j-b+1, \cdots, j$ in B, i.e. intervals ending at i and intervals ending at j are aligned with mapping ratio $a:b$, Δ_w is the difference between the proxy value and interpolated value for w^{th} data point, $\delta_{r,r'}$ is the rate change penalty, i.e. the penalty for changing from a ratio of r to a ratio of r' and $\rho(r)$ is the rate penalty for ratio r, i.e. the penalty for the difference between r and r^*, where r^* is the preferred rate ration, μ is the no match penalty.

c. Termination: Terminate when either sequence reaches the end.

Remark: The initialization $S(i,j,r) = \infty$ for all $i < a$ or $j < b$ is due to the fact that there is no such matching with corresponding ratio and intervals.

2.2 Probabilistic Stratigraphic Model

2.2.1 Modified Pair hidden Markov model

Lisiecki's automated algorithm is a powerful method to find the global optimal alignment, however like other deterministic DPs, it does not yield any information about the uncertainty of an alignment. One approach is to compare different manually aligned alignments [60]. Another disadvantage of the dynamic programming algorithm is that the coefficients for the penalty weightings are selected by trial-and-error and depend on a user's judgement. Thus, all of these motivate us to develop a probabilistic alignment method which is not only able to present the underlying uncertainty, but also estimate the penalty parameters from the data true "automatically".

Our first objective is to develop a pairwise probabilistic alignment algorithm.

The probabilistic algorithm will improve upon the current Dynamic Programming algorithm in the following ways:

1. Maximum likelihood (MLE) or maximum a-posterior parameter (MAP) estimation will allow for automatic parameter estimation based directly on observed paleoclimate data.

2. Sampling of alignments directly from posterior distributions following respective probabilities of each sample alignment.

3. Ensemble-based point estimation of alignments (centroid alignment).

4. Confidence limits on all unknowns at each data point in the alignment, and overall measures of alignment reliability using Bayesian confidence limits.

These improvements will have many applications in paleoclimate research. First, improved parameterizations will improve the accuracy of alignments and also avoid manual trials. Additionally, it is more likely to identify the errors in the optimal solution due to the availability of uncertainty estimates and comparison with other possible alignments. Last but not least, with uncertainty analysis, users will be able to judge whether the alignment is reliable overall and identify regions of an alignment that are less reliable. This is particularly important because usually there is no ground truth for the alignment and the evaluation is often done by comparing paleoclimate proxies from different cores.

Probabilistic alignment algorithms have been extensively used in molecular biology to align pairs of nucleotide and amino acid sequence, and speech recognition in isolated word recognition. These algorithms most frequently employ hidden Markov models, which I have employed here. In our hidden Markov model, the states are composed of a set of plausible mapping rates. In the process of traversing the underlying Markov chain, the alignment sequentially aligns two sequences using different rates. During each mapping, the emission is the sum of the residual square for all the data points and we assume it follows a generalized chi square distribution based on sums of the proxy reading within a interval under the assumptions that the residuals

of these read are distributed $N(0, \sigma^2)$, and the degree of freedom in this chi square distribution is determined by the number of data points falling in these two sets of intervals.

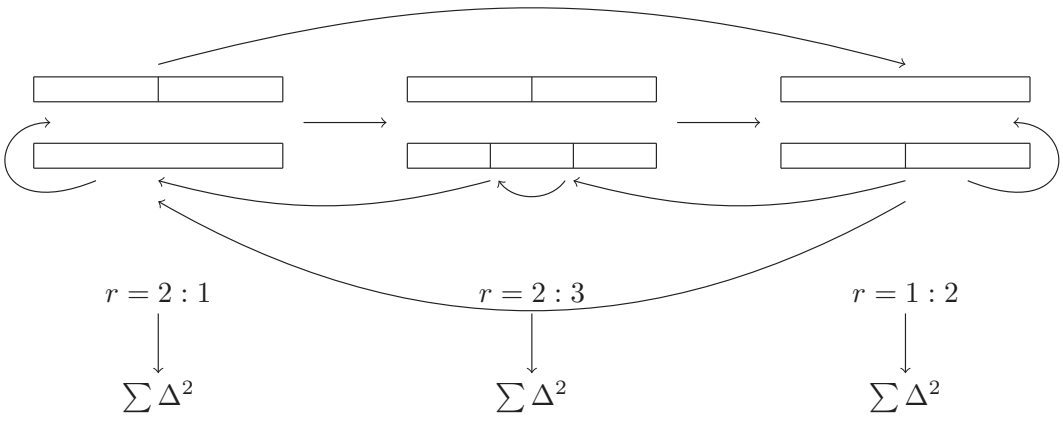


Figure 2.3: hidden Markov model illustration

Figure 2.3 gives the illustration of this pair hidden Markov model. Suppose that we are aligning a pair of sequences, one input sequence and one target sequence. In the general case these two sequences are treated symmetrically. However, in paleoclimate alignment, people usually align the climate records to a stack (a reference record) in order to build the age model.

Notation:

- Input sequence: $Seq_1 = \{u_1, \cdots, u_{N_1}\}$
- Target sequence: $Seq_2 = \{v_1, \cdots, v_{N_2}\}$
- Number of intervals: N_{Int}
- States: $r_1, \cdots, r_{17}$, corresponding to the following mapping ratios

  1:4; 1:3; 2:5; 1:2; 3:5; 2:3; 3:4; 4:5; 1:1; 5:4; 4:3; 3:2; 5:3; 2:1; 5:2; 3:1; 4:1
- Hidden state rate ratio Random Variable: X $= \{X_1, X_2, \cdots, X_t\}$ and $X_i = r_j$ yields the i^{th} mapping ratio.

- Transition probability from state i to state j: $A = (A_{ij})$
- Initial state probability (probability of the starting mapping ratio) : $\pi = \Big(\pi(1), \pi(2), \cdots, \pi(17)\Big)$
- Emissions: $Y_t = (Y_t^1, Y_t^2)$,

  where $Y_t^1 = \{\Delta_{u_p^t}, \cdots, \Delta_{u_q^t}\}$, $Y_t^2 = \{\Delta_{v_{p'}^t}, \cdots, \Delta_{v_{q'}^t}\}$, and $\Delta_{u_i}^t$'s and $\Delta_{v_i}^t$'s are the differences of all of the points within the matched intervals with the linearly interpolated, corresponding values of the other series for the t^{th} mapping. We assume $\sum_{w\in\phi_t} \Delta_w^2 = \sum(Y_t^1).^2 + \sum(Y_t^2).^2 \sim \chi^2(0, \sigma^2, df)$, where ϕ_t denotes the intervals aligned by the t^{th} mapping, $\Delta \sim N(0, \sigma^2)$ and df is the total number of data points falling in these intervals.

Our probabilistic model for stratigraphic alignment is based on the basic pair hidden Markov model. The difference and modification include the following:

1. In the usual hidden Markov models, the observed sequences for the emissions are usually the same for all paths and follow the emission distribution. However, in our setting, the fixed "observation" is the two paleoclimate proxy records which need to get aligned. Instead of the observations, the emission distribution at each state is determined by the difference between these two signals after linear interpolation and thus varies along different mapping paths.

2. The conditional probability distribution of the hidden state variable at time t, given the values of the hidden state variables X at all previous times, depends not only on the value of the hidden state variable X_{t-1}, but actually also depends on all the previous mapping rates since this information is necessary to decide the next intervals to align. In the latter part of this section, we will build this model into a standard "Markov" chain by redefining the hidden variables and thus validates the Markov property of hidden Markov model.

3. We employ the same parameter for the chi square distribution for each state, where the parameter is the standard deviation term σ in the normal distribution $N(0, \sigma)$ of the generalized chi square distribution. Therefore the difference

among different states is the occurrence of the number of data points, i.e. the degree of the chi square distribution.

Specifically, the model can be described as follows. Let us assume for now that we know the distribution on the number of Δ's at each state, i.e. the total number of data points in these mapping intervals, and we know the transition probabilities between the states. A particular alignment can be denoted by a vector $M = \{r_1, r_2, \cdots, r_n\}$, where r_i is the state which represents the mapping rate $r_i = m : n$ and the total number of intervals mapped should be the number of the subintervals in the sequences. During each mapping with rate $r_i = m : n$, it is aligning the next m intervals following the previous mapping in the first sequence with the next n intervals in the second sequence by linear interpolation described in Section **2.1**. There are different possible approaches to deal with the Δ's, which are the differences of all of the points within the matched intervals with the linearly interpolated, corresponding values of the other series. In our work, we choose to use the sum of the squared differences i.e. $\sum \Delta^2$, as the emission at each mapping. We could also let each Δ be one single emission. This is similar to assuming multiple observations per state, as in GHMM.

2.2.2 Validation of the assumption of the hidden Markov model

Before we go further on this model, we should be careful validating the important components of the hidden Markov model in our setting.

1. Markov property of underlying state process

"Markov property" usually refers to the memoryless property of a stochastic process in probability theory and statistics. A stochastic process is said to have the Markov property if the conditional distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it. In our case that the process takes discrete values and is indexed by a discrete time,

this property can be formulated as follows:

$$P(X_n|X_{n-1}, \cdots, X_0) = P(X_n|X_{n-1}).$$

In paleoclimate alignment, it is reasonable to assume that the sediment rate at the current time is only dependent on the sediment rate at the previous time only. Thus the Markov property condition is met and the Markov process is assumed to be homogeneous.

2. Given the state X_n, is the observation Y_n independent of other observations and states?

One difference between this pair hidden Markov model and the usual hidden Markov model is that the observations are not sequentially one-to-one at the time scaling. Instead, depending on the mapping rate, each time we are emitting different number of intervals from each sequence and generating corresponding "observations" Y_n's through linear interpolation. Therefore, the information of the previous state X_n is not enough to deduce the likelihood of observation Y_n, even not enough to "generate" our observation since we need to know the index of intervals in order to make the mapping. Fortunately we could make some minor modification to this model so as to satisfy the independence condition.

For hidden state variable, we use $Z_n = (\sum_{i=1}^{n-1} X_i, X_n)$ instead of X_n. i.e. $Z_1 = (0, X_1), Z_2 = (X_1, X_2), \cdots$. Then Z_n not only stores the information of current mapping rate, but also keeps track of the current starting interval index following the previous mapping. We could also write Z_n as $Z_n = (Z_{n-1}^{(1)} + Z_{n-1}^{(2)}, X_n)$, where $Z_{n-1}^{(1)}$ and $Z_{n-1}^{(2)}$ denote the first and second coordinate of Z_{n-1}. Therefore we have

$$P(Z_n|Z_{n-1}, \cdots, Z_1) = P(Z_n|Z_{n-1}) = I_{\{Z_n^{(1)} = Z_{n-1}^{(1)} + Z_{n-1}^{(2)}\}} P(X_n|X_{n-1}),$$

where I is the indicator function. Then it is clear that this process satisfies Markov property and we assume that the transition probability is homogeneous. Also, given the state $Z_n = (\sum_1^{n-1} X_i, X_n)$, the observation Y_n is independent of other observa-

tions and states.

By introducing the new random variable Z_n, we show that the underlying process satisfies Markov property. For convenience, in the following we still use X_n's as the hidden variable denoting the n^{th} mapping ratio.

3. Is the assumption of normal distribution for Δ's appropriate?

Looking for an appropriate distribution for data is important and sometimes difficult. In this paleoclimate alignment problem, we are looking for optimal alignment which has the smallest sum of squared difference. Therefore we expect small Δ's for good alignment and since we have no preference which sequence would lie above or below the other, it would be reasonable to assume the mean of the Δ's to be zero. From this point of view, normal distribution with mean zero seems to be a good candidate for the following reasons.

(1) The density function for normal distribution is symmetric around mean.

(2) The density decays as the x deviates away from the mean.

(3) The density function of normal distribution with mean zero has the form $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{x^2}{2\sigma^2}}$, where the exponential part has an x^2 term. Then the log of the density would have a x^2 term, which just corresponds to the squared difference error in the dynamic programming algorithm.

Even though normal distribution meet these conditions, we still need to use statistical tools to assure that the underlying data do fit the normal distribution well. In statistics, in order to compare two probability distributions, Q-Q plot is largely used. Q-Q plot plots their quantiles against each other and if the points in the Q-Q plot approximately lie on a straight line, the two distributions being compared are similar. Figure 2.4 gives the Q-Q plot of underlying Δ's from the single optimal alignment of the dynamic programming and random samples from the standard normal distribution. It appears that the Q-Q plot is approximately a straight line which verifies the similar mass distribution of Δ's and normal distribution. Furthermore, this line is roughly centered around $x = 0$, which assures our previous argument of setting the mean of normal distribution zero.

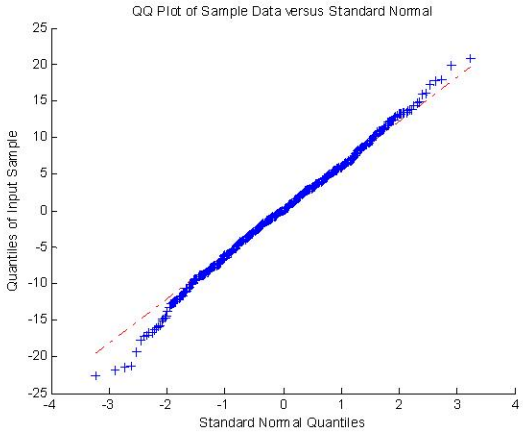


Figure 2.4: **Q-Q plot comparison of the Δ's and normal distribution.** the Q-Q plot approximately lie on a line which verifies the similar mass distribution of Δ's and normal distribution; Furthermore, this line is roughly centered at $x = 0$, which again assures our assumption of mean zero.

In summary, in this section we validate several important features of hidden Markov model to make sure our modeling is appropriate. Then we would be safe to develop the following algorithms to meet the four expectations mentioned earlier in this section.

2.2.3 Viterbi, Forward, Backward Algorithms and parameter estimation

If we assume that the underlying state process is a Markov chain, then the prior probability of state path $M = \{X_1, \cdots, X_n\}$ before observing the data is

$$P(M) = \pi(X_1) \prod_{i=1}^{n-1} A(X_i, X_{i+1}),$$

where $A = (A_{ij})$ is the transition probabilities between states and π is initial distribution. It is often the situation that people would have some prior knowledge or preference over the transitions, e.g. as mentioned in last section, the dynamic programming algorithm chooses a penalty proportional to n_r^2 to more strongly discourage fast changes in accumulation rates. Under this circumstance, it desires to impose a prior distribution on the transition probability A to express user's belief. Thus,

conjugate priors are chosen for the prior distribution on the transition probability A. Specifically, each row of A is multinomial and we choose Dirichlet distribution $Dir(\alpha^i) = (\alpha^i_1, \cdots, \alpha^i_{N_R})$ as prior for each row of A. i.e. $(A_{i1}, \cdots, A_{iN_R}) \sim Dir(\alpha^i)$. Then the likelihood of state path becomes

$$P(A, M) = P(A)P(M|A) = P(A)\pi(X_1)\prod_{i=1}^{n-1} A(X_i, X_{i+1}).$$

Recall that besides transition probabilities A, we still have two parameters, initial distribution π and the standard deviation σ in the chi square distribution. The likelihood actually is

$$f(Seq_1, Seq_2, A, M|\pi, \sigma) = P(A) * f\Big(Seq_1, Seq_2, M|(A, \pi, \sigma)\Big)$$

and

$$\begin{aligned} & f\Big(Seq_1, Seq_2, M|(A, \pi_0, \sigma)\Big) \\ =&\pi(X_1)\prod_{i=1}^{n-1} A(X_i, X_{i+1})\prod_{i=1}^{n} f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i}\Delta^2_w\Big), \end{aligned} \tag{2.1}$$

where $f_{\chi^2(\sigma)}(\cdot)$ is the density function for generalized chi square distribution with underlying normal distribution $N(0, \sigma^2)$.

1. Forward algorithm: the full likelihood, summing over all paths

The full likelihood of Seq_1 and Seq_2 is calculated by summing over all possible alignments:

$$f\Big(Seq_1, Seq_2|(A, \pi, \sigma)\Big) = \sum_{\text{alignment M}} f\Big(Seq_1, Seq_2, M|(A, \pi, \sigma)\Big),$$

To simplify the notation, in this section we omit the conditioning information (A, π_0, σ) when we write the likelihoods, e.g. $f(Seq_1, Seq_2|(A, \pi, \sigma))$ is written as

$f(Seq_1, Seq_2)$ for short. Let

$$f(i,j,r) = \sum_{\text{alignment M}} f\Big(Seq_{1\{1:i\}}, Seq_{2\{1:j\}}, M = (X_1, \cdots, X_t) | X_t = r\Big)$$

be the combined likelihood of all alignments up to i^{th} interval of Seq_1 and j^{th} interval of Seq_2 that end in mapping ratio $r = a : b$. i.e. intervals $i - a + 1, \cdots, i$ in Seq_1 is aligned with intervals $j - b + 1, \cdots, j$ in Seq_2. Let $\sum_{w \in \phi^r_{i,j}} \Delta^2_w$ be sum of the squared difference between proxy and their interpolation. Given the condition of the last mapping ratio $r = a : b$, the mapping before the last one must end at $i - a$ and $j - b$. And $f(i,j,r)$ is independent of the information contained in the dynamic table $f(1 : i - a - 1, 1 : j - b - 1, :)$, since the full likelihood up to $(i - a, j - b)$ is summarized by $f(i - a, j - b, :)$. So we have the following recursion.

a. Initialization: For every possible ratio $r = a : b$, set

   i. $f(i,j,r) = 0$ for all $i < a$ and $j < b$

   ii. $f(i,b,r) = f\Big(\sum_{w \in \phi^r_{i,b}} \Delta^2_w\Big)\pi(r)\pi_0^{i-a}$ for all $i \geq a$

   iii. $f(a,j,r) = f\Big(\sum_{w \in \phi^r_{a,j}} \Delta^2_w\Big)\pi(r)\pi_0^{j-b}$ for all $j \geq b$

b. Recursion:

$$f(i,j,r) = f\Big(\sum_{w \in \phi^r_{i,j}} \Delta^2_w\Big) \sum_{r' \in R} \Big(A_{r',r} f(i - a, j - b, r')\Big)$$

c. Termination:

$$f^E(N_{\text{int}}, N_{\text{int}}) = \sum_{r \in R} f(N_{\text{int}}, N_{\text{int}}, r)$$

Then, the full likelihood is

$$f(Seq_1, Seq_2) = f^E(N_{\text{int}}, N_{\text{int}}).$$

An important use of the full likelihood is to define a posterior distribution $P(M|Seq_1, Seq_2)$ over alignment M given a pair of sequence Seq_1, Seq_2, which is

given by

$$P(M|Seq_1, Seq_2) = \frac{f(Seq_1, Seq_2, M)}{f(Seq_1, Seq_2)}.$$

If we set $M = M^*$, the most probable alignment, we are able to obtain the posterior probability of the most probable alignment $\frac{v^E(N_{\text{int}},N_{\text{int}})}{f^E(N_{\text{int}},N_{\text{int}})}$. This posterior probability could be interpreted as the probability that the optimal scoring alignment is "correct" in our model.

2. Viterbi algorithm: the most probable path

If we are interested in a single solution or estimate, the maximal likelihood estimate (MLE) or the maximum a posterior estimate (MAP) are most frequently used. In our hidden Markov model setting, MLE and MAP estimate can be easily obtained by recursion via dynamic programming. We let

$$V(i, j, r) = \max_{\text{alignment M}} f\Big(Seq_{1\{1:i\}}, Seq_{2\{1:j\}}, M = (X_1, \cdots, X_t)|X_t = r\Big)$$

be the likelihood of the most probable state path up to i^{th} interval of Seq_1 and j^{th} interval of Seq_2 provided that the last mapping ratio is $r = a : b$. Then we obtain similar recursion as in forward algorithm:

a. Initialization: For every possible ratio $r = a : b$, set

   i. $V(i, j, r) = 0$ for all $i < a$ and $j < b$

   ii. $V(i, b, r) = f\Big(\sum_{w\in\phi^r_{i,b}} \Delta^2_w\Big)\pi(r)\pi_0^{i-a}$ for all $i \geq a$

   iii. $V(a, j, r) = f\Big(\sum_{w\in\phi^r_{a,j}} \Delta^2_w\Big)\pi(r)\pi_0^{j-b}$ for all $j \geq b$

b. Recursion:

$$V(i, j, r) = f\Big(\sum_{w\in\phi^r_{i,j}} \Delta^2_w\Big) \max_{r'\in R} \Big(A_{r',r}V(i - a, j - b, r')\Big)$$

c. Termination:

$$V^E(N_{\text{int}}, N_{\text{int}}) = \max_{r'\in R} V(N_{\text{int}}, N_{\text{int}}, r')$$

and the optimal alignment path can be found by backtracing. First, let $\omega_1 = N_{\text{int}}, \omega_2 = N_{\text{int}}$ and the last state is determined by

$$X_t = \arg\max_{r \in R} V(\omega_1, \omega_2, r).$$

Setting $\omega_1 = \omega_1 - a, \omega_2 = \omega_2 - b$, then the previous state could be found recursively:

$$X_s = \arg\max_{r \in R} V(\omega_1, \omega_2, r) A(r, X_{s+1}) f\Big(\sum_{w \in \phi^r_{\omega_1,\omega_2}} \Delta_w^2\Big).$$

Note that the probability $A(r, X_{s+1}) f\Big(\sum_{w \in \phi^r_{\omega_1,\omega_2}} \Delta_w^2\Big)$ used in the recursion is exactly what we used when sampling back.

3. Backward sampling algorithm: probabilistic sampling of alignments

In order to generate a sample alignment, we trace back through the matrix of $f(i, j, r)$ values. Instead of taking the highest scoring as the choice at each step, probabilistic choice is chosen based on the relative strengths of different components. To illustrate, imagine that we are in the middle of the traceback, in state ratio $r = a : b$ until position (i, j), we know from the forward algorithm that

$$f(i, j, r) = f\Big(\sum_{w \in \phi^r_{i,j}} \Delta_w^2\Big) \sum_{r' \in R} \Big(A_{r',r} f(i - a, j - b, r')\Big).$$

Until the beginning of either sequence, we choose the next step to be r' with probability:

$$\frac{f\Big(\sum_{w \in \phi^r_{i,j}} \Delta_w^2\Big) A_{r',r} f(i - a, j - b, r')}{f(i, j, r)}. \tag{2.2}$$

Drawing one sample only takes linear time as can be seen from Equ. (2.2) and hence it is very convenient and practical to draw samples from the posterior space. Furthermore, posterior samples are very useful in inference and estimates, e.g. the samples can be used to approximate the marginal probability of interval i in Seq_1 aligned with interval j in Seq_2. We can simply count the fraction of samples that has interval i aligned to interval j. In the exact formula to estimate this marginal probability,

another "backward recursion" will be required. Thus the sampling approach is much easier for implementation. We should remind the reader that this estimate based on posterior samples only reflects the inference uncertainty under this particular model we are working with. Anyone who does not have enough confidence in the model itself should be very cautious when analyzing the result, even when the underlying uncertainty is zero.

4. Baum-Welch algorithm : parameter estimation

There is no direct closed form estimate for the estimation of parameters when the paths are unknown for sequences. A particular iteration method, Baum-Welch algorithm [9] is standardly used, which is a special case of the Expectation Maximization (EM) algorithm. Informally, Baum-Welch first estimates the expected number of transitions with respect to $P(\pi|x, \theta^t)$, then use these expected numbers to obtain new values of A_{ij}'s. This process is iterated until some stopping criterion is reached.

Our algorithm of parameter estimation is based on Baum-Welch algorithm, however with slight modification since our model differs from the standard HMM. First, we put a prior distribution on the transition probabilities A, and therefore the estimate (optimal solution) we are looking for is the MAP estimate instead of the ML estimate.

$$\hat{\theta}_{ML}(x) = \arg\max_{\theta} f(x|\theta)$$

$$\hat{\theta}_{MAP}(x) = \arg\max_{\theta} f(x|\theta)P(\theta)$$

The optimization of MAP estimate has one more term, the prior probability of the parameter, than the ML estimate. Thus we should modify the EM algorithm in order to get the posterior likelihood increased at every iteration. Second, the emission probability is continuous instead of discrete. Then we could not "count" the expected number of emission of certain symbols. The procedure is essentially based on Durbin et al. [31].

Now we aim to find the model that maximizes the log likelihood

$$\log f\Big(Seq_1, Seq_2|(A,\pi,\sigma)\Big)+\log P(\theta) = \sum_M \log f\Big(Seq_1, Seq_2, M|(A,\pi,\sigma)\Big)+\log P(\theta). \tag{2.3}$$

The missing data is the alignment M. Let us use notation $\theta = (A, \pi, \sigma)$. Then the $Q(\theta|\theta^t)$ function in the EM algorithm Becomes

$$Q(\theta|\theta^t) = \sum_M f(M|Seq_1, Seq_2, \theta^t)\Big(\log f(Seq_1, Seq_2, M|\theta) + \log P(A)\Big). \tag{2.4}$$

For a given path each parameters of π and A in the model will appear some number of times in $f(Seq_1, Seq_2, M|\theta)$ given by the product (2.1). To simplify the notation, we incorporate the initial probability into the transition probability A by adding one row to the transition probability $(A_{01}, \cdots, A_{0N_R})$ in the beginning. Thus under this notation row 0 of A is the initial probability. Now for transition probability we call this number $\tilde{A}_{kl}(M)$, which is the number of times the state transits from k to l. Then we can rewrite (2.1) as

$$p(Seq_1, Seq_2, M|\theta) = \Big[\prod_{k=0}^{N_R}\prod_{l=1}^{N_R} A_{kl}^{\tilde{A}_{kl}(M)}\Big] \times \prod_{i=1}^{t} f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i^M}\Delta_w^2\Big). \tag{2.5}$$

After taking the logarithm, (2.4) can be written as

$$\begin{aligned} Q(\theta|\theta^t) = &\sum_M f(M|Seq_1, Seq_2, \theta^t)\times \\ &\Big[\sum_{k=0}^{N_R}\sum_{l=1}^{N_R}\tilde{A}_{kl}(M)\log(A_{kl}) + \sum_i \log\left(f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i^M}\Delta_w^2\Big) + \log P(A)\right)\Big]. \end{aligned} \tag{2.6}$$

Let $\tilde{A}_{kl} = \sum_M p(M|Seq_1, Seq_2, \theta^t)\tilde{A}_{kl}(M)$, then $\tilde{A}_{kl}$ is the expected number of transitions from state k to l with respect to $p(M|Seq_1, Seq_2, \theta^t)$. Doing the sum over M

first in (2.6) and replacing the prior distribution on A with

$$P(A) = \Big(\frac{1}{B(\alpha)}\Big)^{N_R+1} \prod_{k=0}^{N_R}\prod_{l=1}^{N_R} A_{kl}^{\alpha_{kl}-1},$$

therefore it gives

$$\begin{aligned} Q(\theta|\theta^t) = &\sum_{k=0}^{N_R}\sum_{l=1}^{N_R}(\tilde{A}_{kl}+\alpha_{kl}-1)\log(A_{kl}) + \\ &\sum_{M}\log f(M|Seq_1, Seq_2, \theta^t)\sum_{i}\log\left(f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i^M}\Delta_w^2\Big)\right). \end{aligned} \tag{2.7}$$

Finally we need to maximize $Q(\theta|\theta^t)$, i.e. the M-step. Let us first take a look at the A term. The difference between this term for $A_{ij}^* = \frac{\tilde{A}_{ij}+\alpha_{ij}-1}{\sum_k(\tilde{A}_{ik}+\alpha_{ik}-1)}$ and for any other A_{ij} is

$$\sum_{k=0}^{N_R}\sum_{l=1}^{N_R}\tilde{A}_{kl}\log\frac{\tilde{A}_{kl}^*}{\tilde{A}_{kl}} = \sum_{k=0}^{N_R}\Big(\sum_{l'}\tilde{A}_{kl'}\Big)\sum_{l=1}^{M}\tilde{A}_{kl}^*\log\frac{\tilde{A}_{kl}^*}{\tilde{A}_{kl}}.$$

The last expression is actually the relative entropy between A^* and A, and thus it is always larger than zero unless $A_{kl} = A_{kl}^*$. This shows that it achieves maximum at A_{kl}^*. We have

$$A_{ij}^* = \frac{\tilde{A}_{ij}+\alpha_{ij}-1}{\sum_k(\tilde{A}_{ik}+\alpha_{ik}-1)}. \tag{2.8}$$

Now we turn to the second term in (2.7), which involves the σ term. We are not able to write it in such an easy form as the first term involving A. However, this term is just the expected sum of the logarithm likelihood for the $\Big(\sum\Delta^2\Big)$'s during each interval mapping. Even for A_{ij} terms, we get an explicit form of the MAP estimate $A_{ij}^* = \frac{\tilde{A}_{ij}}{\sum_k\tilde{A}_{ik}}$. Although it is possible, it is still complicated to derive the $\tilde{A}_{ij}$'s, i.e. the expected number of transitions. This is where the samplings become quite useful. To avoid the problem of having to deal with the task of summing over all possible alignments, we use a variants of EM called stochastic EM.

5. Stochastic EM

The difference between stochastic EM and ordinary EM is that we use samples

from the distribution $P(M|x,\theta^{old})$ to approximate the summation in $Q(\theta|\theta^t)$. Actually it is like Monte Carlo method and we replace the summation in (2.4) with samples drawn from the posterior using parameters obtained from the last iteration:

$$Q(\theta|\theta^t) = \frac{1}{N}\sum_{n=1}^{N}\Big(\log f(Seq_1, Seq_2, M^{(n)}|\theta) + \log P(A)\Big),$$

where $\{M^{(n)}\}_1^N$ are samples from $f(M|Seq_1, Seq_2, \theta^{old})$. However, now we could not guarantee that the likelihood would get increased after each iteration and thus could no longer guarantee to obtain a (local) maximum of the likelihood. Additionally, we are not certain that the algorithm will converge, although it generally works well in practice with a large number of representative samples. Now we just take derivative with respect to the A_{ij}'s, it is straightforward to derive that

$$A_{ij}^{\star} = \frac{\hat{A}_{ij} + \alpha_{ij} - 1}{\sum_k(\hat{A}_{ik} + \alpha_{ik} - 1)}, \tag{2.9}$$

where $\hat{A}_{ij} = \frac{1}{N}\sum_{i=1}^{N}\tilde{A}_{ij}(M^{(n)})$, the averaged number of transitions in the N samples. Comparing the estimate from Baum-Welch algorithm (2.8) and that from stochastic EM (2.9), they are equivalent in that $\hat{A}_{ij}$ is just the Monte Carlo estimate of $\tilde{A}_{ij}$.

For σ term, we just need to look for the σ which maximizes the sum of term $\sum_i \log\Big(f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i^{M^{(n)}}}\Delta_w^2\Big)\Big)$ for each alignment $M^{(n)}$.

In summary, our procedures to estimate the parameters are as follows:

Until convergence

{

i. *Build the forward table* $f(i,j,r)$ *using* θ^{old}.

ii. *Draw* N *samples from the posterior distribution* $P(M|Seq_1, Seq_2, \theta^{old})$.

iii. *Update* A_{ij}*'s and* σ. *Specifically, update* A_{ij}*'s as in (2.9) and choose* σ *to maximize* $\sum_{n=1}^{N}\sum_i \log\Big(f_{\chi^2(\sigma)}\Big(\sum_{w\in\phi_i^{M^{(n)}}}\Delta_w^2\Big)\Big)$.

}

2.2.4 Yet, another parametric model

Although the model above looks all good, we still face the problem of choosing the appropriate weight for the Dirichlet prior distribution, i.e. how many pseudo count shall we put during the iterative estimation. Due to the fact that there is no ground truth, we have no way to judge the correctness of the alignment result. As different pseudo count would definitely generate different estimates, it becomes difficult to decide the appropriate value for the weight of the prior. Notice that in the extreme case where the pseudo count is high, the resulting transition probabilities would just follow the prior distribution. From this point of view, it seems like the choice of pseudo count should depend on how we believe on the prior distribution we choose. Another problem with the model above is over fitting due to the large number of unknown parameters (mostly come from the 17×17 transition matrix) and relatively small number of observatative data. We have also tried to implement hierarchical model by using multiple sequences as training data and use empirical Bayesian to derive the optimal parameter for the Dirichlet distribution. However, the pseudo count turn out to be small and make almost no effect in the following estimation. This phenomenon might be due to the non similarity among the training data and hence it desires different parameters in different alignments.

Recall that the motivation of our employing Dirichlet prior is that in Lisiecki's dynamic programming algorithm, it desires to strongly discourage the fast changes in cumulation rate. In order to still incorporate this prior preference, we modify the model a bit, actually in a simpler way, which employs a parametric form for the transition probabilities. The two important penalties in the dynamic programming algorithm are rate penalty and rate change penalty. We impose the parametric form for the transition probabilities as

$$A_{ij} = \frac{\exp\Big(- C_\rho(\tilde{r_j} - 1)^2 - C_\delta(i-j)^2 \Big)}{Z_i}, \tag{2.10}$$

where $\tilde{r} = \frac{1}{r}$ if $r < 1$ and $\tilde{r} = r$ if $r \geq 1$, and Z_i is the normalization constant for

the i^{th} row of the transition matrix A. In (2.10), we adopt the penalty terms in Lisiecki's dynamic programming algorithm and use it as the Boltzmann weight in the transition probabilities. By doing so, lower penalty would correspond to higher Boltzmann weight and therefore lead to higher probability for this term.

With the parametric form (2.10), the degree of freedom of A reduces from 17×17 to only 2. Furthermore, in addition to avoid the determination of the choice of pseudo count, we also avoid the potential problem of over fitting. Corresponding algorithms for this simplified model is similar with those in Section 2.2.3. During the iterative EM step, instead of updating A using (2.9), we should choose the parameters C_ρ and C_δ that maximize the empirical likelihood.

2.3 Application to the alignment of $\delta^{18}O$ (oxygen isotope ratio) records and uncertainty analysis

Centroid estimator In addition to the most probable alignment, centroid alignment, the posterior risk minimizer is also investigated. As shown in [14], the posterior Hamming loss risk minimizer $\hat{\theta}_C(S)$ is actually the posterior marginal sum maximizer:

$$\hat{\theta}_C(S) = \arg\max_{\hat{\theta}\in\Theta} \sum_{i=1}^{n} P(\theta_i = \hat{\theta}_i(S)|S). \tag{2.11}$$

Carvalho [14] also proved that $\hat{\theta}_C$ is the posterior p^{th} power loss minimizer and $\hat{\theta}_C$ minimizes the squared distance to the posterior mean. Therefore this estimator is the nearest feasible solution to the center of mass of the posterior space. Moreover, $\hat{\theta}_C$ represents the central tendency of the space since it also depends on the distance to other points and their probabilities. Therefore, it has quite different behavior from the MLE and MAP estimate: the centroid estimator seeks to minimize the expected posterior-weighted distance instead of choosing a single peak likelihood in the space. We could also prove that the centroid estimator $\hat{\theta}_C$ is equivalent to the

median estimator for binary variables, therefore there are at least half ensemble lying above it and at least half ensemble below it. The proof is quite straightforward.

Proof of $\hat{\theta}_C$ is the median: For binary variables, $(\hat{\theta}_C)_i$ takes the value which has probability larger than 0.5. The median estimator is defined to be the numerical value separating the higher half of a sample from the lower half, i.e. we choose median to be α which satisfies $P(X \leq \alpha) \geq \frac{1}{2}$ and $P(X \geq \alpha) \geq \frac{1}{2}$. For binary variables, if we assume that $P(X = 1) \geq \frac{1}{2}$, then we have $P(X \leq 1) = P(X = 0) + P(X = 1) = 1 \geq \frac{1}{2}$ and $P(X \geq 1) = P(X = 1) \geq \frac{1}{2}$. Therefore, the median estimator is $\hat{X} = 1$, which is the same as the centroid estimator.

Example 1: δ^{18}O proxy sequence from ODP site 980 in North Atlantic

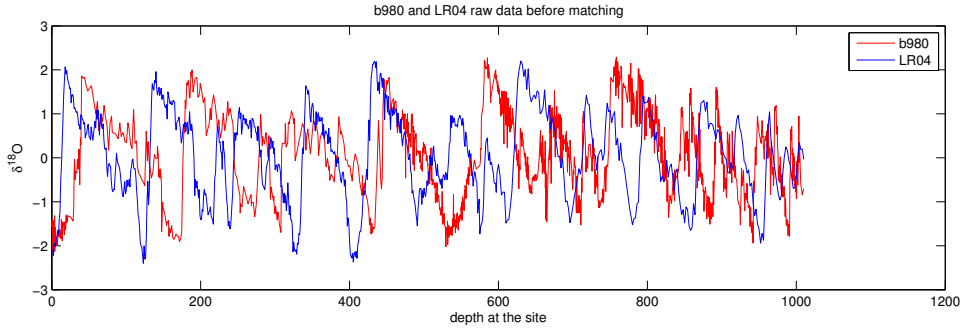


Figure 2.5: **b980 and LR04 raw data before matching.** The input sequence $b980norm$ is the δ^{18}O (oxygen isotope ratio)from the calcium carbonate shells of benthic foraminifera from Ocean Drilling Program Site 980 in the North Atlantic.The target sequence $LR04normsh$ is a stack (average) of δ^{18}O records from 57 different ocean sediment cores from around the world.

Figure 2.5 gives the input and target sequences before alignment. The input sequence $b980norm$ is the δ^{18}O (oxygen isotope ratio) from the calcium carbonate shells of benthic foraminifera from Ocean Drilling Program Site 980 in the North Atlantic. It is a measure of both the deep ocean temperature at that location and the size of continental ice sheets (ice volume). The x-axis is depth in meters in the sediment core (i.e., meters below the sea floor). The target sequence $LR04normsh$ is a stack (average) of δ^{18}O records from 57 different ocean sediment cores from

around the world. Therefore, it is a globally averaged record of ice volume and deep ocean temperature. Lisiecki et al. [55] employed orbital tuning to infer ages for to the points in the $LR04normsh$ stack and that permits the assignment of ages to the x-axis in thousands of years ago (ka). Figure 2.6 gives the alignment result from Lisiecki's MATCH software. The upper figure is matching alignment and the lower figure reflects corresponding matching "speed" of the alignment. A speed greater than one means that the signal is compressed in that part of the alignment, and a speed less than one means that the signal is stretched in the alignment. As comparison, Figure 2.7 gives the optimal Viterbi alignment, i.e. the most probable alignment from hidden Markov model.

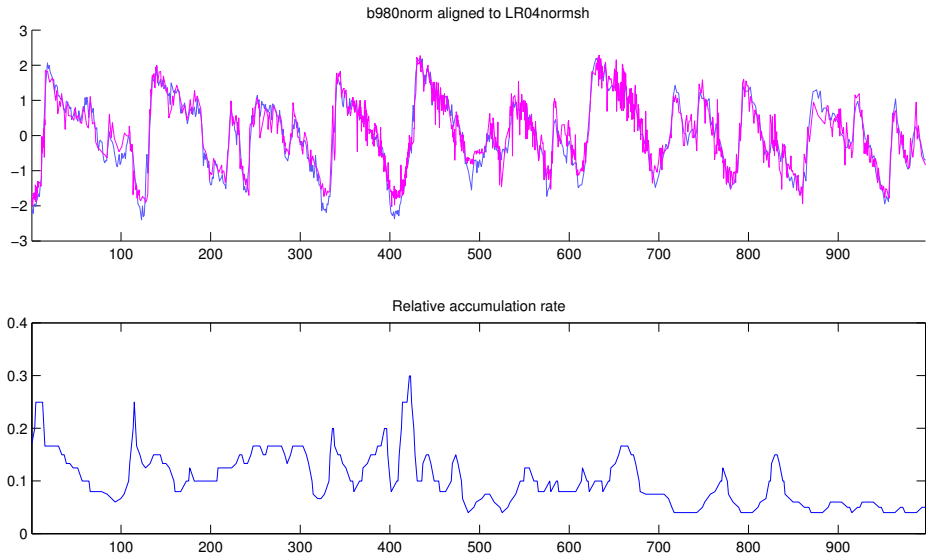


Figure 2.6: b980norm aligned with LR04normsh using MATCH

Comparing Figure 2.6 and Figure 2.7, we could see that the alignments from Match and hidden Markov model are very close to each other, and their corresponding aligned aging is given in Figure 2.9. For the accumulation rate, although the rate changes more frequently in Viterbi alignment, the main pattern is still similar with that of Match alignment.

Δ's reflect the difference of the linear interpolation between the input sequence

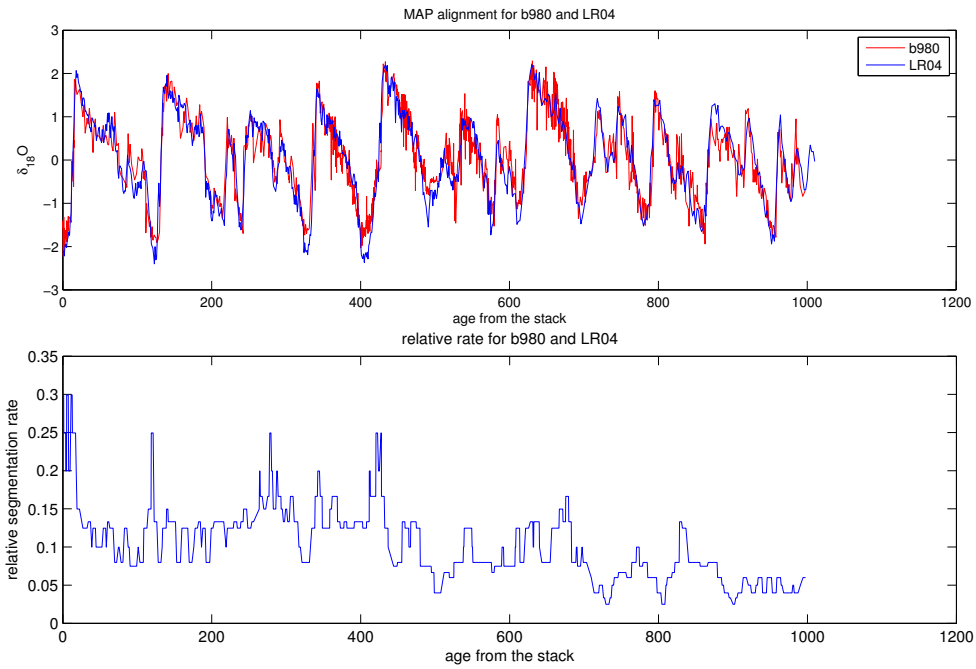


Figure 2.7: **b980 aligned with LR04 using HMM.** The upper figure is the most probable (Viterbi) alignment in HMM model; The lower figure reflects corresponding matching "speed" of the alignment.

and target sequence. As pointed out in Section 2.2, one advantage of employing probabilistic model is that we are able to draw independent samples following their probability distribution. Therefore, among the sample alignments, the mean of Δ's tells us how much on average the input sequence deviated from the target sequence, while the standard deviation of Δ's provides us the spread of the Δ's among the sample alignments. Figure 2.8 gives the mean and standard deviation of the Δ's in 1000 sampled alignments. As comparison, the standard deviation obtained from EM algorithm is 0.4009, which seems like a good match in explaining the spread in Figure 2.8. On the other hand, after observing the mean of the Δ's, we found that the Δ's are indeed distributed around zero, which fits our previous argument of setting the mean of the normal distribution zero.

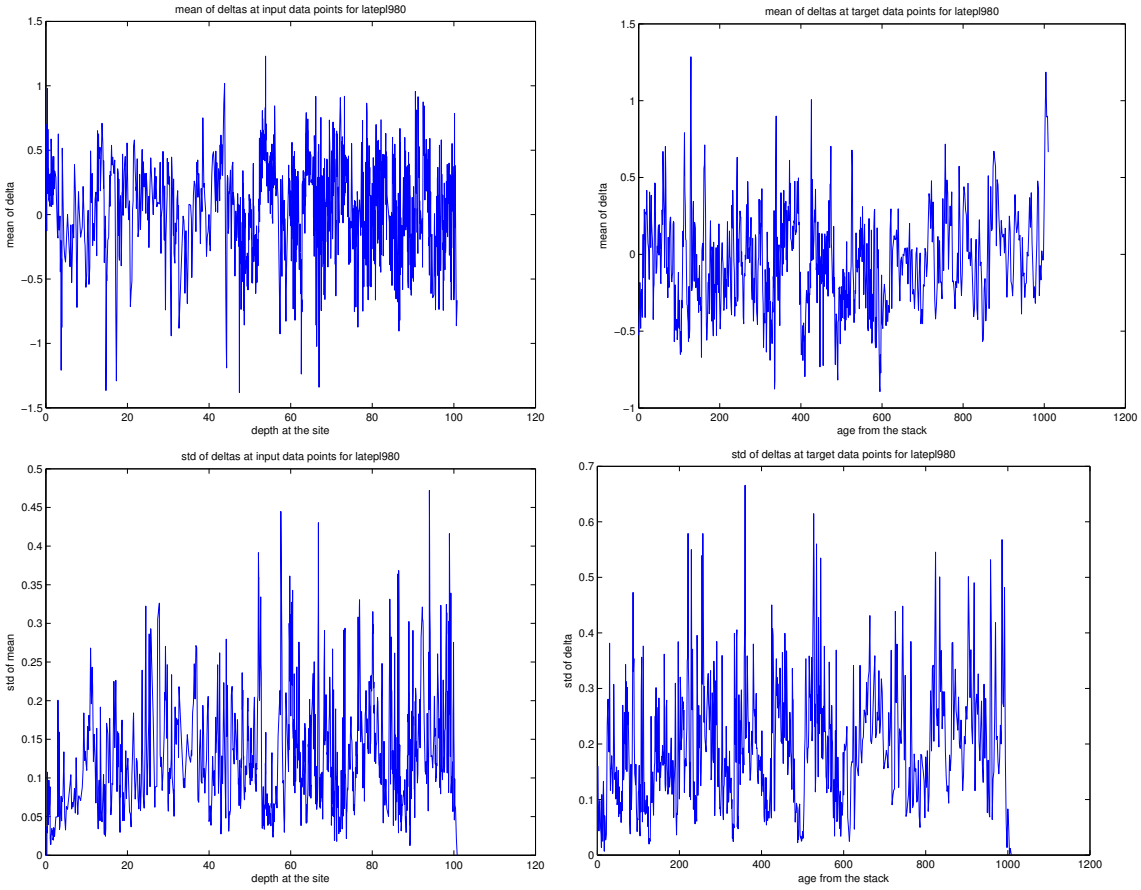
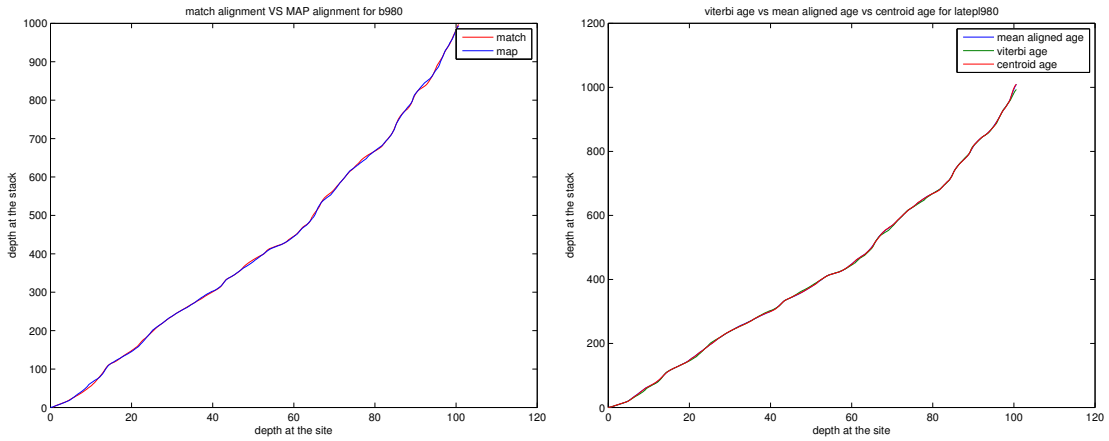


Figure 2.8: mean and std of delta's from input and target sequence for b980

Furthermore, we are also able to build the alignment between the depth of these

two sites. Figure 2.9 compares various alignments, including Match alignment, mean aligned depth, Viterbi alignment and centroid alignment. Centroid alignment has the smallest quadratic difference to the mean alignment and presents the tendency of the whole space. The depth alignment allows users to look up an depth from the stack for each depth in their core. As the curves show, the difference between these alignment are very small. As shown in Figure 2.10, the 95% Bayesian credibility band in the aging curve are generally quite tight (less than 5), which means the alignment result should be quite reliable at most sites. However, the aging is noticeable less certain at some other sites, having standard deviation over 10. From this point of view, users could make their analysis of aging especially cautious for the sites with high uncertainty. Again, we need to point out that all the uncertainty analysis is based on our belief on the correctness of this model.



(a) (b)

Figure 2.9: **Comparison of various alignment.** (a) Match alignment compared with Viterbi alignment; (b) Comparison of mean alignment, Viterbi alignment and centroid alignment

In Figure 2.10, it is also clear that the Viterbi alignment is very close to the centroid alignment in this example. Therefore, both could be chosen as the representative or optimal alignment when a single alignment is required. For the 95% credibility band, the patterns at most sites are very similar except some minor difference at some sites, although the credibility band is usually tighter for centroid

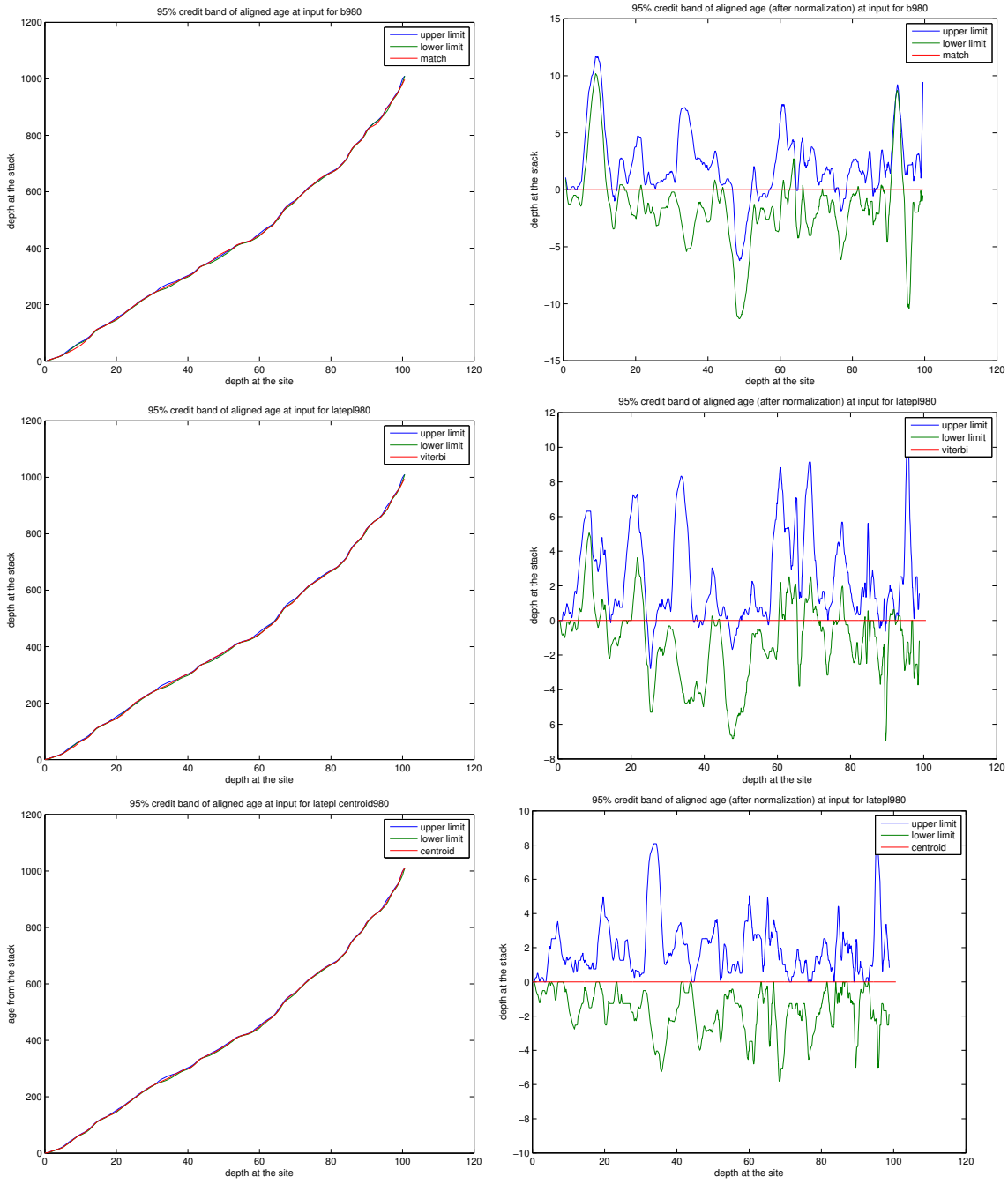


Figure 2.10: 95% credibility band and scaled 95% credibility band of the aligned depth for input seq b980

alignment. Since the centroid alignment returns the median of the sum overall intervals it does enjoy the advantage of tending to track the center of the posterior mass. This advantage explains why the confidence limits are tighter around it than around the MAP alignments. Also notice that in a few places the MAP alignment falls a little outside of the confidence limits.

Example 2: δ^{18}O proxy sequence from ODP site 980 in North Atlantic

In this example, we display the same set of plots as that in Example 1. Figure 2.11 gives the raw data for input and target sequence and Figure 2.12 is the alignment result from MATCH. Compared with Figure 2.6, the alignment of sequence la1089 does not match the target sequence as well as that of sequence b980 in that the difference between sequences is apparently much more noticeable. In this sense, it is reasonable for us to expect that the Δ's in this example might be higher than that in the previous example. In addition, we notice that the accumulation rates change more frequently in the Viterbi alignment, which might due to the less favorable of transition into the same state in the resulting estimate of the transition matrix A. However, this is what the data tells us based on this model. If users are unsatisfied with that, they might choose to impose some prior distribution.

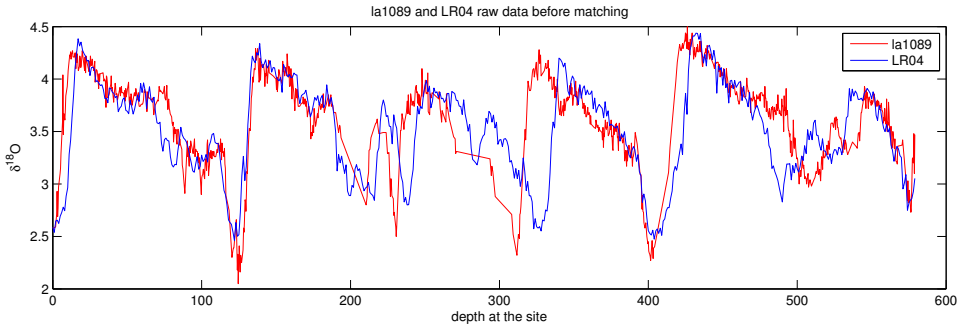


Figure 2.11: la1089 and LR04 raw data before matching

Again, the means of Δ's still lie around zero and the standard deviation of the Δ's is smaller than that in Example 1. As comparison, the standard deviation σ we obtain from EM algorithm is 0.1160.

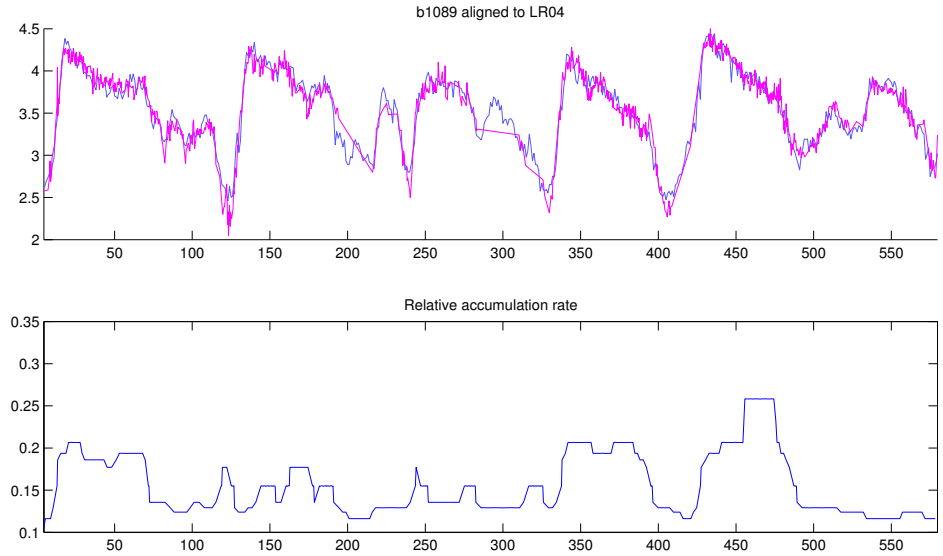


Figure 2.12: la1089 aligned with LR04normsh using MATCH

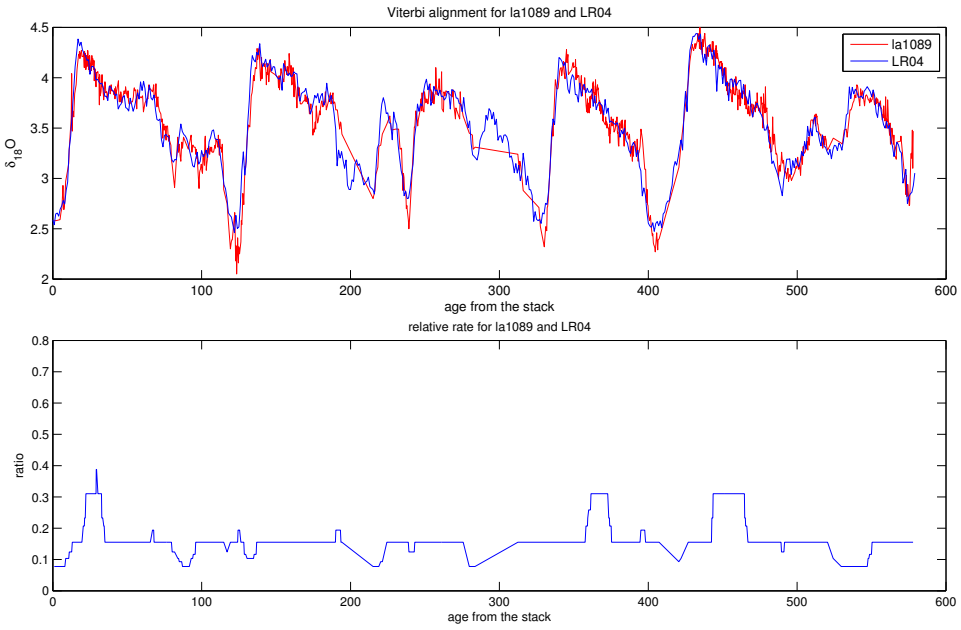


Figure 2.13: **la1089 aligned with LR04 using HMM.** The upper figure is the most probable (Viterbi) alignment in HMM model; The lower figure reflects corresponding matching "speed" of the alignment.

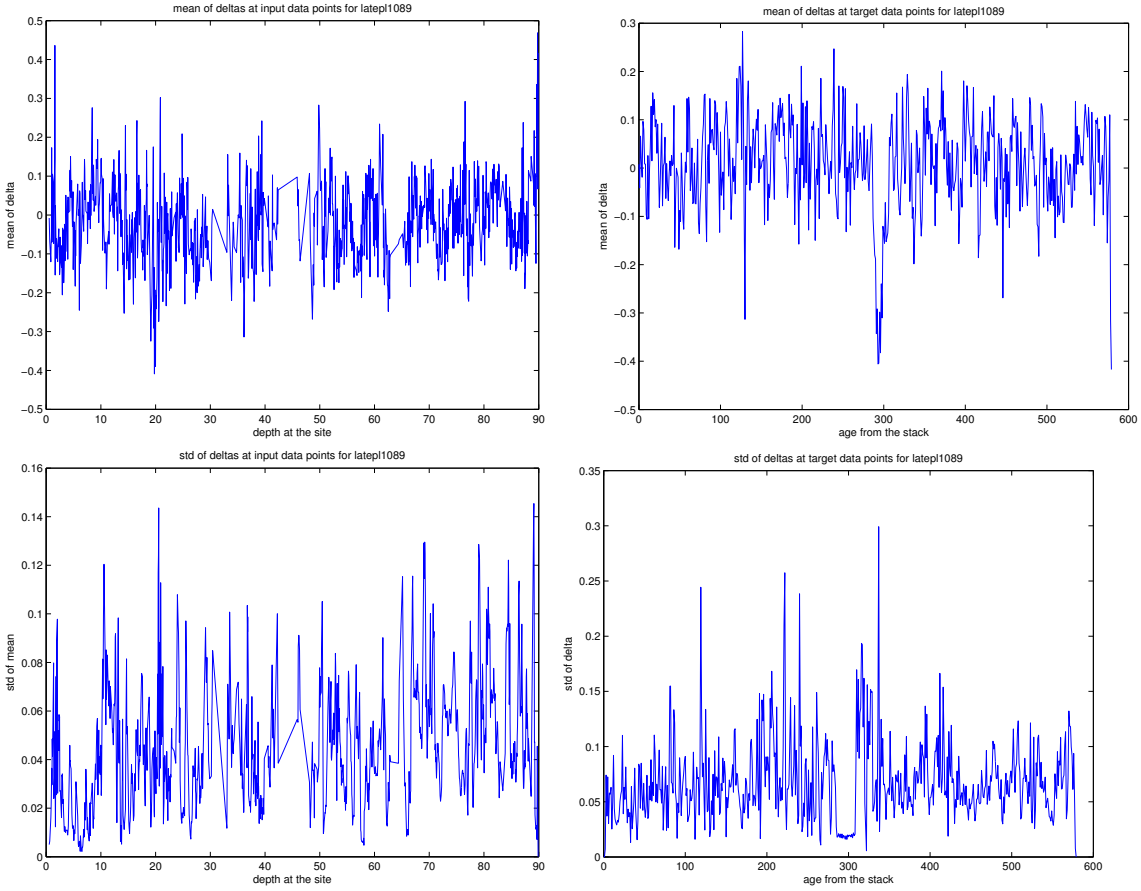
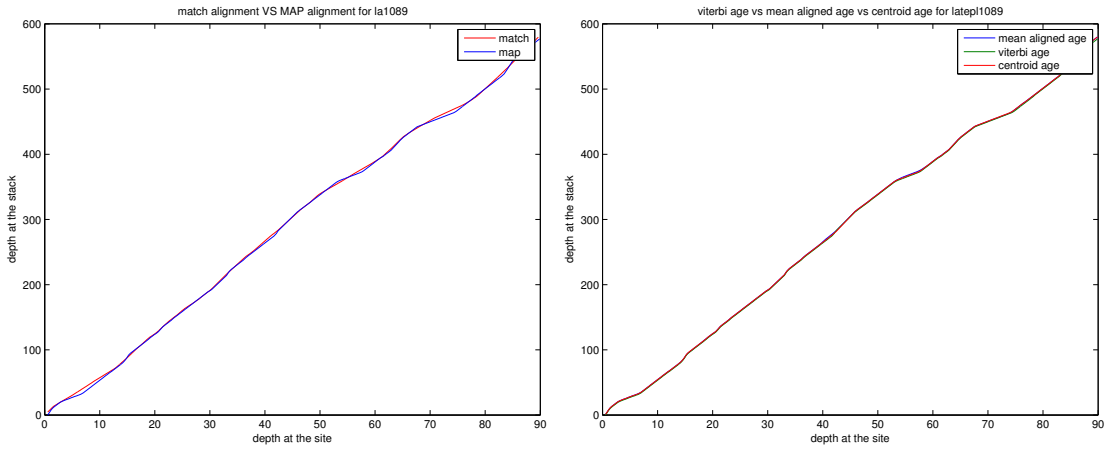


Figure 2.14: mean and std of delta's from input and target sequence for la1089

In this example, Figure 2.15 represents difference between the MATCH alignment and the Viterbi alignment. They have slight difference from each other at the beginning of the core. Also, the mean alignment, Viterbi alignment and centroid alignment also differ slightly at some sites. Furthermore, in Figure 2.16 we could see that the credibility band for centroid alignment is a bit tighter than the mean Viterbi alignment. This is again an illustration that the centroid alignment is the one represents the global tendency. The most striking aspect of this alignment is that the MAP alignment is very often below the lower confidence limits and is also above the upper confidence limit in several places. On the other hand there is no such deviation for the centroid alignment. Again the centroid has the advantage of being pulled to the center of the data because it is the overall median alignment.



(a) (b)

Figure 2.15: **Comparison of various alignment for la1089.** (a) Match alignment compared with Viterbi alignment; (b) Comparison of mean alignment, Viterbi alignment and centroid alignment

2.4 Discussion and Future work

In this chapter, we develop a probabilistic pair hidden Markov model for stratigraphic alignment based on Lisiecki's deterministic dynamic programming algorithm. We validate the assumption of this hidden Markov model and provide algorithms to

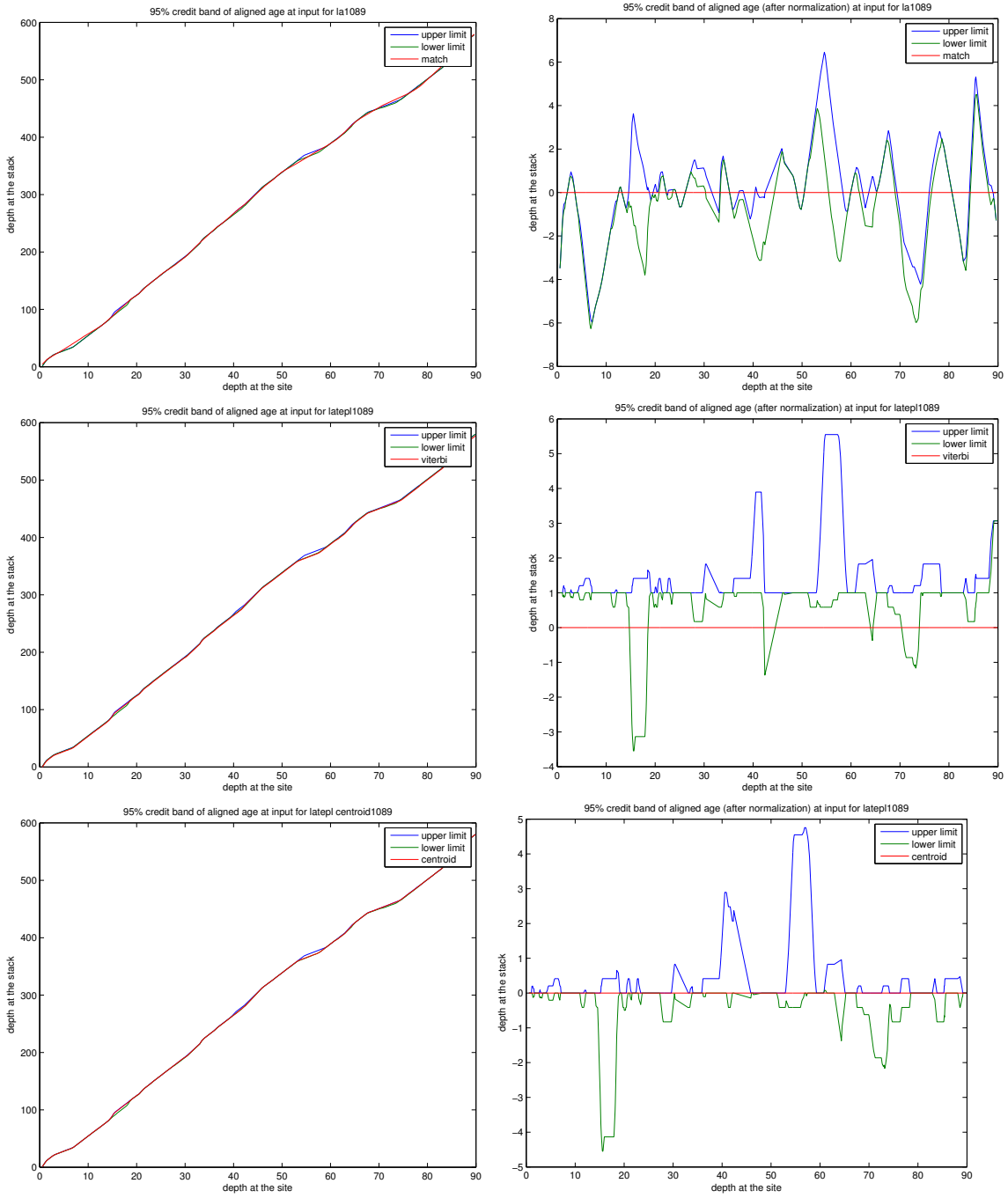


Figure 2.16: 95% credibility band and scaled 95% credibility bands of the aligned depth for input seq la1089

obtain Viterbi alignment, centroid alignment, along with corresponding uncertainty analysis. Due to the fact that centroid alignment presents the central tendency, and centroid alignment is actually the median estimator and often has tighter confidence band than MLE, centroid alignment is very important when considering the candidate for a single optimal solution in addition to Viterbi alignment.

However, there exist several limitations in the DP algorithm. First, the block size needs to be selected appropriately, smaller for high resolution data and larger for low resolution data in order to cover more data in each subinterval. Second, the mapping rates are expected to be higher for the data which is sparse. Third, the linear interpolation between records allows for no error for between points, which leaves no degree of freedom. However, it is clear we would have less uncertainty if the points are close to each other and apparently much higher uncertainty for sparse data.

Based on these limitations, we would continue work on improvement upon the current components of this pair hidden Markov model. First, the interval mapping via a set of plausible accumulation rates might be changed in a cunning way that continuous mapping or point based mapping is available. Second, instead of linear interpolation, we might explore some other forms of periodic functions with higher order (quadratic or spine) to get estimates of values for between points. Last but not least, instead of following Lisiecki's direction as the parametric form in transition probability matrix, a more appropriate form or method to incorporate this prior knowledge might be explored.

Furthermore, in paleoclimatology research, multiple alignments among sequences is also desirable and is important in building the global model. In hidden Markov model, multiple alignment is implemented as profile HMM, which is commonly achieved in molecular biology and speech recognition. Our next goal therefore is to build a probabilistic profile HMM stratigraphic alignment model based on the work of Lisiecki & Raymo [56].

Part III

Characterization of High-D Space

Chapter 3

Mutual Information based Iterative algorithm to Characterize High-D space

In high-dimensional spaces, the common clustering algorithms mentioned in the introduction are impractical since we are unable to enumerate the whole space. Subspace clustering is one solution to high dimensional setting in reducing the dimensionality. One similar work is feature selection. In machine learning and statistics, feature selection (variable selection or feature reduction) selects a subset of features from a large number of features to characterize the data. However in order to achieve optimal feature selection for supervised learning, it requires an exhaustive search of all subsets of these features. This is usually impractical if the number of features is large. Hence in practice people usually only search for a satisfactory set of features instead of searching for the real optimal set. However, feature selection does not always give satisfactory results, especially when the underlying space is particularly complex or the dimension is high. This leads to the question when does feature selection work? Existence of exclusive constraints might be one of the situations. Exclusive constraints excludes the possibility of values of some variables given the value of some certain variables, therefore simplifying the space a lot.

What is a good or optimal characterization? In supervised situation it often

means to minimize the classification error, while in unsupervised situation, it usually requires the maximal dependency of the target cluster on the data in corresponding subspace. This scheme is maximal dependency. Mutual information has been used in feature selection as criteria of max-dependency, max-relevance and min-redundancy [68].

In information theory, entropy is a measure of the uncertainty associated with a random variable, while the mutual information of two random variables X and Y measures the mutual dependence of the two random variables, i.e. the amount of information shared between these two random variables. Coin tossing is the typical example to illustrate entropy in which we are interested in the entropy of a single toss of a not necessarily fair coin. If the coin is fair, it has an entropy of one bit and this is the situation in which it is most difficult to predict the outcome of next tossing since each side has equal probability to appear. However, if the coin is not fair, one side is more likely to appear than the other side and then the uncertainty (entropy) is lower. The extreme case is the probability of head up being 1 or 0, in which situation there is no uncertainty on the outcome of next tossing and the entropy is zero.

Intuitively, mutual information measures the information shared between X and Y: it measures how much uncertainty of the other variable would be reduced if we know the information of one of these variables. For example, if X and Y are independent with each other, then knowing X does not give any further information about Y and vice versa, so their mutual information is zero. At the other extreme, if X could be totally determined by Y or vice versa, the mutual information is the maximal possible uncertainty in X or Y. In this sense, mutual information actually measures statistical dependence. The concept was introduced by Shanan (1948). In a wide variety of applications, people may want to maximize mutual information or often equivalently, to minimize the conditional entropy.

In this chapter we construct a characterization criteria based on entropy measure and propose an algorithm which identifies the most informative variables as an avenue for improved characterization of the high dimensional space. We iteratively

look for the most informative variable by comparing the averaged pairwise mutual information which measures the information one certain variable tells about the rest and use different states of these variables to describe the probabilistic models of different subspaces. Therefore it could be implemented as an unsupervised clustering procedure. In addition, the advantage of this algorithm is that we are now able to derive the distribution on the subspaces by conditional probability and characterize the important features of each subspace by a small number of informative variables. Furthermore, we are able to draw independent samples from subspaces and make statistical analysis based on these samples. The method is proposed and illustrated in detail in Section 3.1. As an application, in Section 3.2 we illustrate this algorithm by the example of the posterior space of RNA secondary structure.

3.1 Looking for informative variables - recursive algorithm

Dependency analysis are very basic in statistics. And it is commonly evaluated by regression analysis and correlation analysis. Using correlation to measure the strength of statistical dependency is long standing, however, in many situations mutual information has advantages over it. Mutual information measures not only linear dependency like correlation, but also reveals some other relationships. Also, the variables involved need not to be euclidean in mutual information analysis. The idea behind our algorithm is to look for the variables with high dependency with other variables, i.e. the most informative variables which could tell the most of other variables, and afterwards we are able to partition the posterior space based on the states of these informative variables. We employ mutual information as the criteria of the dependency between random variables.

Before we introduce the method, we specify the notations which will be used in this section.

Let us assume that $X = (X_1, X_2, \cdots, X_d)$ is the d dimensional random variable

in the probability space Σ, where d is large. We draw N independent samples from this space, $X^{(1)}, X^{(2)}, \cdots, X^{(N)}$ following their probabilities (densities), along with their corresponding probabilities $p_1, p_2, \cdots, p_N$ (or their proportion subject to a normalization constant). We also assume that the sample size N is sufficiently large to guarantee statistical reproducibility in typical sampling statistics.

Definition and Equivalence:

1. Let S be a collection of indices (e.g. S=$\{1, 2, 3, 4\}$).
2. Let $X_{s \in S}$ be the projection of the random variable X to the s^{th} coordinate.
3. X_A, where $A \subseteq S$ is defined as $\{X_s : s \in A\}$.
4. ${}_B X^A$ is defined as $X_{A \backslash B} = \{X_s : s \in A \backslash B\}$.

3.1.1 Mathematical Formulation

Intuition of idea For a certain one dimensional random variable X_i projected from $X = (X_1, \cdots, X_d)$, we would like to consider the dependency of this random variable X_i and other random variables. We randomly select another random variable X_j, where $j \neq i$. Thus (j, X_j) is a random pair with respect to random variable X_i. We tend to measure the amount of information X_i shares with the remaining random variables, i.e. how much information X_i could tell about the others, or what could we tell about the other variables if we have the information of X_i. One of the most popular approaches to realize Max-Dependency is maximal relevance. Relevance is usually characterized in terms of correlation or mutual information, of which the latter is one of the widely used measures to define dependency of variables. In our work we use the mutual-information-based measure $I\Big(X_i; (j, X_j)\Big)$, which is the mutual information between X_i and the corresponding random pair (j, X_j). For this

measure $I\Big(X_i;(j,X_j)\Big)$, we have

$$\begin{aligned} I\Big(X_i;(j,X_j)\Big) &= H(X_i) - H\Big(X_i|(j,X_j)\Big) \\ &= H(X_i) - E_X\Big[\log\frac{P\Big((j,X_j)\Big)}{P\Big(X_i,(j,X_j)\Big)}\Big] \\ &= H(X_i) - \frac{1}{d-1}\sum_{j\neq i} E_X\Big[\log\frac{P(X_j)}{P(X_i,X_j)}\Big] \\ &= H(X_i) - \frac{1}{d-1}\sum_{j\neq i} H(X_i|X_j) \\ &= \frac{1}{d-1}\sum_{j\neq i}\Big(H(X_i) - H(X_i|X_j)\Big) \\ &= \frac{1}{d-1}\sum_{j\neq i} I(X_i;X_j). \end{aligned} \tag{3.1}$$

As illustrated in (3.1), $I\Big(X_i;(j,X_j)\Big)$ is actually just the averaged pairwise mutual information between X_i and all other different X_j's. In other words, $I\Big(X_i;(j,X_j)\Big)$ is a measurement of the averaged information X_i shares with the remaining variables ${}_jX$. This understanding of $I\Big(X_i;(j,X_j)\Big)$ is very natural and straightforward. Once we realize this, in order to look for the most informative variable, it is natural that we select the variable based on maximal relevance criterion (Max-Relevance):

$$\max_i I\Big(X_i;(j,X_j)\Big), \tag{3.2}$$

where $I\Big(X_i;(j,X_j)\Big) = \frac{1}{d-1}\sum_{j\neq i} I(X_i;X_j)$. In (3.2), we calculate the averaged pairwise mutual information between one certain variable and the remaining variables, and choose the one with the highest dependency as the most informative variable. Employing basic mathematical calculation, we have the following equivalence:

$$\sum_j H(X_j) = H(X_i) + \sum_{j\neq i} I(X_i;\ X_j) + \sum_{j\neq i} H(X_j\ |X_i). \tag{3.3}$$

From (3.3), the sum of entropy for all the random variables is equivalent to the sum of the uncertainty (entropy) of one certain random variable X_i, the information of X_i shares with all the other random variables (mutual information), and the uncertainty of other random variables given the information of X_i (conditional entropy). In (3.2) we are maximizing the averaged pairwise mutual information between X_i and other variables, which is equivalent to maximize the second term on the R.H.S of (3.3). One might also choose to minimize the uncertainty of other random variables given the information of X_i. Both are reasonable criteria for clustering analysis. Actually from the relationship in (3.3), it is easy to see that:

$$\max_i \sum_{j \neq i} I(X_i;\ X_j) = \min_i \Big(\sum_{j \neq i} H(X_j\ |X_i) + H(X_i), \Big)$$

$$\min_i \sum_{j \neq i} H(X_j\ |X_i) = \max_i \Big(\sum_{j \neq i} I(X_i;\ X_j) + H(X_i). \Big)$$

These two optimization only differ in a single term $H(X_j)$. In high-dimensional space, the number of terms in the summation is very large, therefore this single term $H(X_i)$ might make little contribution in the optimization. Hence, the results from these two optimization scheme usually should not be far away from each other especially in high-dimensional spaces. In our work, we choose to maximize the mutual information term as the criteria to select the most informative random variable.

So far, we are able to find the most informative variable from the optimization in (3.2), which then could help us to describe the subspaces partitioned by different states of this most informative random variable. Our procedure could be iterative, i.e. depending on the stopping criteria, e.g. how fine we would like to partition each subspaces (or how many clusters shall we define), we could iteratively look for the next most informative variables within them. The recursive algorithm is described as below.

Algorithm 3.1.1: *Iteratively looking for most informative variables based on maximum averaged pairwise mutual information.*

1. *Look for the most informative variable by*

$$s^{(1)} = \max_i I\Big(X_i; (j, X_j)\Big), \;\; I\Big(X_i; (j, X_j)\Big) = \frac{1}{d-1} \sum_{j \neq i} I(X_i; X_j). \tag{3.4}$$

2. *Look for the most informative variables within each subspaces defined by different states of* $X_{s^{(1)}}$*:* $\Omega_t^{(1)} = \{X : X_{s^{(1)}} = x_{s^{(1)}{}_t}\}$.

$$s_t^{(2)} = \max_{i: i \neq s^{(1)}} I\Big(X_i; (j, X_j) | X_{s^{(1)}} = x_{s^{(1)}{}_t}\Big). \tag{3.5}$$

3. *Repeat step 2 in respective sub-subspaces until meeting stopping criteria.*

Setting stopping criteria is usually an important step. Our goal here is to divide the space into characteristic subspaces and we would have two extreme cases in which we might not want to go further. Sometimes it would end with some subspace within which the elements are all similar to each other, while sometimes it may end with some subspace within which the elements are all quite different from each other. In information theory terms, the first situation corresponds to $I(X_i; X_j) = H(X_i) - H(X_i|X_j) \approx H(X_i) \approx 0$. The first approximation is due to the fact that all elements are similar to each other and then given the information of X_j the uncertainty of another elements would be quite small, i.e. $H(X_i|X_j) \approx 0$. On the other hand, since all elements are similar to each other, the overall uncertainty is small and we have $H(X_i) \approx 0$. For the other extreme case, the situation is opposite. Since the subspace is complicated enough that none of these variables is informative about others and we would have $I(X_i; X_j) = 0$. Therefore, both extreme situations would lead to the quantity that $I(X_i; X_j) \approx 0$. From this point of view, we could use the information measure as the stopping criteria and stop when the average pairwise mutual information is close to zero.

Extra Constraints and some considerations There might be several considerations before implementing this algorithm. First, we don't want the cluster found to be too large or too small. If the probability of the cluster is too small, all the

ensembles in it might be considered as outliers and would be of no interest to us. As the first step in this algorithm, we use different states of the most informative variable to partition the space into different subspaces. While we require the mass proportion of each cluster to be no less than a threshold ϵ, it is equivalent to restrict the variable candidates such that

$$\epsilon \leq P(X_i = x_s) \leq 1 - \epsilon.$$

Now that we tend to use the most informative variable to give clusters to the high-dimensional space, a more straightforward measure would come out to be $H({}_iX|X_i)$, which gives the uncertainty of the rest given the information of X_i. However, when we write out the maximization scheme, we would find that this measure is not practical. Employing basic mathematics, we have $H({}_iX|X_i) = H(X) - H(X_i)$. i.e. given the total entropy $H(X)$ a fixed constant, minimizing $H({}_iX|X_i)$ is equivalent to maximizing $H(X_i)$, which achieves maxima when probability distribution over X_i is equally likely. Performing this scheme would find those variables with equal probability for each states as the winner and partition the posterior space into similar sized subspaces. Under some circumstances, partitioning the space into similar size subspaces might be a good scheme, while it is not the case for most of the situations, e.g. the posterior space of RNA secondary structure. In the posterior space of RNA secondary structure, we have already seen a lot of examples whose bimodal posterior space are with non-close proportions, e.g. 0.8 to 0.2 or 0.9 to 0.1. Therefore, $H({}_iX|X_i)$ is not a good goal measure in this example.

Conversely, how about $H(X_i|{}_iX)$? $H(X_i|{}_iX)$ is the uncertainty of X_i given the information of the rest. If random variable X_i is indeed a good candidate to differentiate a multi-modal high-dimensional space, knowing states of all other variables should provide us a good prediction for the state of X_i. This condition therefore could be a pre-requisite for the most informative variable candidate. We also need to notice that this condition could only be a pre-requisite, not the determinant condition. Given the dimension d large, the dimension for the conditioning variable ${}_iX$ is much

higher than the one dimensional variable X_i. With such huge difference and very possible dependencies between different variables, it is highly likely not difficult to predict X_i when we have all the rest information. Once there is one variable among the rest that has high dependency with X_i, the conditional uncertainty $H(X_i|_iX)$ becomes small. This is also a reason we choose to maximize the pairwise mutual information instead of just maximizing $I(X_i;\ _iX)$, which looks more natural at the first glance. We have

$$I(X_i;\ _iX) = H(X_i) - H(X_i|_iX).$$

Form the argument above, in high-dimensional space once there exists a random variable X_s that has high dependency with X_i, $H(X_i|_iX)$ would become very small. In this case, $I(X_i;\ _iX)$ achieves its maximum possible value $H(X_i)$ and is very likely to be selected as the most informative random variable. In particular in the posterior space of RNA secondary structure with exclusive constraints, given the states of all other base pairings, often that we are able to deduce the paring state of this particular base pair due to this exclusive constraint that there is no pseudoknot (base pair crossing) in the structure. In this situation, comparing $I(X_i;\ _iX)$ is approximately comparing $H(X_i)$, which is not practical as discussed before. In all, under such circumstance, the dependency between X_i and another variable X_s masks the relation between X_i and other random variables. One single dependency dominates as the whole dependency. Alternatively if we use the averaged pairwise mutual information, this single dependency only contributes to one term of the summation and we would still be able to take the relationship between X_i and other random variables into account.

In summary, two possible constraints user might hope to impose to this algorithm are:

$$\epsilon \leq P(X_i = x_s) \leq 1 - \epsilon, \tag{3.6}$$

$$H(X_i|_iX) \leq \delta. \tag{3.7}$$

Employing constraints may help the optimization search for the right candidates we need. It could also help to get fewer candidates and thus save time complexity which might be high specially in high dimensional space.

3.1.2 Computation and Time Complexity

For convenience, we assume that in the d-dimensional space, the space for each dimension is Ω with cardinality $|\Omega|$.

Calculation of $I(X_i; X_j)$: First, from the formula for mutual information, we have

$$I(X_i; X_j) = \sum_{x_i \in \Omega} \sum_{x_j \in \Omega} P_{X_i,X_j}(x_i, x_j) \log \frac{P(x_i, x_j)}{P_{X_i}(x_i) P_{X_j}(x_j)}. \tag{3.8}$$

In order to compute (3.8), we need to marginalize the joint probability P_X to get P_{X_i,X_j}, P_{X_i} and P_{X_j} :

$$P_{X_i,X_j}(x_i, x_j) = \sum_{{}_{\{i,j\}}X} P(x_i, x_j, {}_{\{i,j\}}X), \tag{3.9}$$

$$P_{X_i}(x_i) = \sum_{{}_iX} P(x_i, {}_iX). \tag{3.10}$$

Apparently the computational complexity for the direct calculation of (3.9) and (3.10) are $|\Omega|^{d-2}$ and $|\Omega|^{d-1}$ respectively. Therefore the complexity for $I(X_i; X_j)$ as in (3.8) is $|\Omega|^{d-1}|\Omega| + |\Omega|^{d-2}|\Omega|^2 = \mathrm{O}(|\Omega|^d)$. In order to do the optimization in (3.4), we need to compute the mutual information for every pair of variables (X_i, X_j), hence the total complexity would be $\mathrm{O}(M|\Omega|^d)$, where M is number of candidate pairs of variables.

Calculation of $I(X_i; {}_iX)$:

$$I(X_i; {}_iX) = \sum_{X_i} \sum_{{}_iX} P(X_i, {}_iX) \log \frac{P(x_i, {}_ix)}{P_{X_i}(x_i) P_{{}_iX}({}_ix)} \tag{3.11}$$

From the same argument as above, for a specific $x_i \in X_i$ and ${}_ix \in_i X$, the marginal

probability $P_{X_i}(x_i)$ and $P_{iX}({}_ix)$ require computational complexity of $\mathrm{O}(|\Omega|^{d-1})$ and $\mathrm{O}(|\Omega|)$ respectively. Therefore, the total complexity for (3.11) would be $\mathrm{O}(|\Omega|^{d-1})\mathrm{O}(|\Omega|)$ + $\mathrm{O}(|\Omega|)\mathrm{O}(|\Omega|^{d-1}) = \mathrm{O}(|\Omega|^d)$.

In high dimensional space, complexity of $\mathrm{O}(|\Omega|^d)$ is obviously too expensive. If X is a Markov Random Field, we might decrease the complexity through dynamic programming, which will be shown in the RNA secondary structure example in the following section. In this specific example, we provide an exact calculation algorithm which decreases the complexity to $\mathrm{O}(N^5)$ where N is the length of the RNA sequence. However, this is still not practical enough. Thereby we also provide a sampling based approach in the next section.

3.2 Application to computational biology

3.2.1 Background of RNA secondary structure

In this section we illustrate the algorithm with specific application to computational biology.

RNA molecules are essential elements in some of the cell's most fundamental processes. The function of a structural RNA molecule is determined by its structure to a large degree. Prediction of RNA secondary structure is one of the most fundamental problems in computational biology. In many cases where the crystal structures are not available, computational methods for modeling RNA secondary structure have been proven to be valuable. Historically people usually regard the searching for the "optimal" secondary structure for a RNA sequence as an optimization problem and the minimum free energy structure was sought [85].

Free energy minimization is long established in computational biology. Free energy minimization is based on the assumption that the underlying molecular folding at equilibrium is unique and the molecule would fold into the lowest energy state. Free energy minimization strategy has been applied in RNA folding [85], transmembrane helix packing [67] and protein folding [3, 1]. Specifically, it has been widely

applied to the prediction of RNA secondary structure with good success.

Besides the MFE perspective, McCaskill [62] extended a probabilistic interpretation to build probabilistic Boltzmann weighted free energy model. Dynamic programming could be implemented to calculate the partition function over the ensemble of structures (under some constraints) given a RNA sequence. Following McCaskill, Ding et al. [30] developed a stochastic back-trace algorithm which is able to draw samples from the Boltzmann weighted ensemble of secondary structures from this free energy model. Another probabilistic approach is to employ a stochastic context-free grammar [31]. Regardless of the probabilistic model, the joint distribution of a RNA sequence X and its secondary structure A could always be represented as

$$P(X, A) \propto \exp\{-\frac{1}{T}E(X, A)\},$$

where T is the parameter and E is the free energy of X with secondary structure A.

Ding et al. [29] showed evidence of multimodal posterior ensembles that there often exist several distinct clusters in the Boltzmann ensemble. The structures within each cluster are similar to each other. To aid in the characterization of the sampled structural space, [27] introduced the centroid structure as an efficient means to characterize the central tendency for a set of structures. Carvalho and Lawrence [14] proved that centroid estimator is the posterior Hamming loss risk minimizer and centroid estimator minimizes the square distance to the posterior mean. Furthermore, when the clusters within these spaces are well separated, it is not likely that any single solution, including the centroid, is able to represent the posterior space well. In such cases, it is desirable to use multiple centroids to represent the space [27, 29]. By drawing a set of statistical samples of RNA secondary structure from the Boltzmann weighted ensemble, Ding et al. [29] described a procedure to capture the main features of the space. With these representative samples, this procedure does clustering on the samples and obtain centroids for each cluster.

In current work, clusters are usually obtained based on the samples drawn from the posterior space. For hierarchical clustering of structure, Sfold [18] employed

Diana [48], a top-down divisive method that has performed well in other contexts [24]. In this section, we will perform Algorithm **3.1.1** in Section 3.1 to look for the most informative base pairs to characterize the posterior space and give clusters based on them.

Let us first formalize the posterior space of RNA secondary structure. The observed data X is a RNA sequence consisting of n nucleotide bases. We represent a secondary structure A in our space by a binary matrix of indicator variables : A_{ij}, with $i = 1, \cdots, n-2$ and $j = i+2, \cdots, n$, $A_{ij} = 1$ if base i in the X is paired to base j and $A_{ij} = 0$ otherwise (consecutive positions do not form pairs). The constraint is that each base could only be paired with at most one other base, and so we require that $\sum_{i=1}^{n-2} A_{ij} \leq 1$ for $j = i+2, \cdots, n$ and $\sum_{j=i+2}^{n} A_{ij} \leq 1$ for $i = 1, \cdots, n-2$. Even with these constraints, there still exists an exponentially large number of possible structures. To permit the use of a polynomial time algorithms for this problem, one can impose a set of exclusivity constraints know as no pseudo knot constraints. No-pseudoknot constraints of the form $A_{ij} + A_{kl} \leq 1$ with $1 \leq i < k < j < l \leq n$ are introduced, that is, a pseudoknot occurs when both $< i, j >$ and $< k, l >$ are paired. In sum, the constraints could be written in terms of random variables A_{ij} as below:

1. $\sum_{i=1}^{n-2} A_{ij} \leq 1$ for $j = i+2, \cdots, n$;
2. $\sum_{j=i+2}^{n} A_{ij} \leq 1$ for $i = 1, \cdots, n-2$;
3. $A_{ij} + A_{kl} \leq 1$ for $1 \leq i < k < j < l \leq n$.

Circle representation is usually used to visualize these exclusivity constraints. In this circle we draw chords to pair base i and base j if $A_{ij} = 1$. Then, we could judge whether the underlying structure violates the exclusivity constraint easily by observing the circle representation. If two chords have a common endpoint, then they violate the constraints of at most one pairing. If two chords cross each other, then they violate the no-pseudoknot constraint. Figure 3.1 illustrates the pairing diagram and the circle diagram to represent the exclusivity constraints.

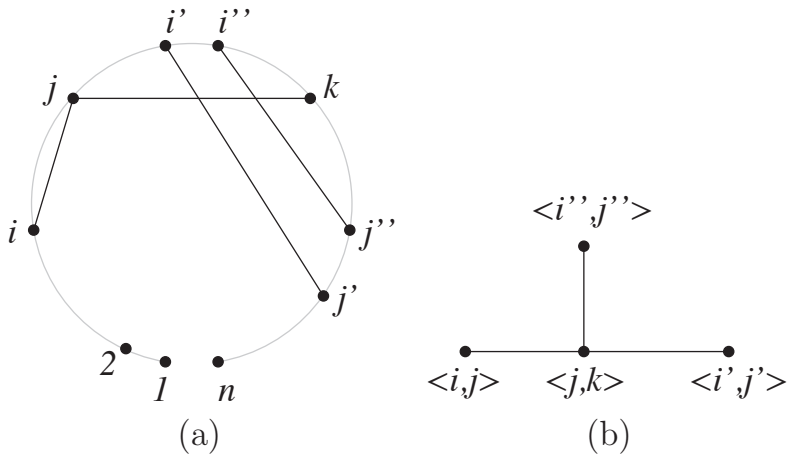


Figure 3.1: **RNA secondary stricture.** (a) Pairing diagram. (b) Constraint circle graph. Figure reproduce from Carvalho's thesis [15]

3.2.2 Exact and empirical calculation

To demonstrate the exact calculation of mutual information, we present in detail an algorithm for the stacking energy model using dynamic programming described in [26]. We denote the sequence from i^{th} base to j^{th} base of an RNA molecule of n ribonucleotides by $R_{ij} = r_i r_{i+1} \cdots r_j$, $1 \leq i, j \leq n$. Let SU_{ij} denote a secondary structure on this subsequence and this secondary structure meets both the no pseudoknot constraint and at most one pairing constraint. Additionally, we use SP_{ij} to denote the structures of the subsequence R_{ij} with base i paired with base j. Then the partition function restricted to R_{ij} is defined as:

$$u(i,j) = \sum_{SU_{ij}} \exp[-E(R_{ij}, SU_{ij})/RT], \tag{3.12}$$

$$up(i,j) = \sum_{SP_{ij}} \exp[-E(R_{ij}, SP_{ij})/RT], \tag{3.13}$$

where $E(R_{ij}, SU_{ij})$ is the free energy for the secondary structure SU_{ij}, R is the gas constant, T is the absolute temperature, and $kcal/mol/RT = 1.6625$. In order to derive recursions with stacking energies, we divide the cases into three mutually

exclusive cases for R_{ij}.

1. Nucleotide r_j is single stranded;

2. Nucleotide r_j is paired with nucleotide r_i;

3. Nucleotide r_j is paired with another nucleotide r_k, where $i < k < j$.

Then we could write the recursions for the conditional partition functions as follows:

$$u(i,j) = u(i,j-1) + up(i,j) + \sum_{i<k<j} u(i,k-1)up(k,j), \tag{3.14}$$

$$up(i,j) = \exp(-E([i \cdot j/(i+1)] \cdot [(j-1)/RT])up(i+1,j-1)), \tag{3.15}$$

where $E([i \cdot j/(i+1)] \cdot [(j-1)/RT])$ is the stacking energy between the adjacent base pair $r_i - r_j$ and $r_{i+1} - r_{j-1}$. In (3.14), the three components in $u(i,j)$ correspond to the three mutually exclusive cases mentioned above. Then, by this recursion, we could obtain the partition function for the whole sequence R_{1n} by dynamic programming.

In the application of the posterior space of RNA secondary structure, we are interested in characteristic base pairs. Due to the exclusivity and the pairing to at most one other base constraints, the formation of some base pairs would prohibit the formation of some other base pairs to a large degree. From the clustering examples of messenger RNAs [29], the centroid of each cluster usually contains different stems (a set of continuous base pair stacking) from each other and these stems often cross upon each other, thus violating the exclusivity constraints, i.e. under our secondary structure model, a structure could not have these stems at the same time. This inspires us to use some base pairs to characterize and distinguish different clusters. Based on the dynamic programming algorithm above, we provide here an exact computation to the optimization algorithm in (3.4), in which the goal is to identify the most informative base pairs. In order to optimize over all base pairs to look for the one that maximizes the average pairwise mutual information, we need to build a pairwise mutual information table for $I(A_{ij}; A_{i'j'})$.

Optimization of average pairwise mutual information McCaskill [62] developed a dynamic programming algorithm that can calculate the partition function for secondary structure formation in $\mathrm{O}(n^3)$ time and $\mathrm{O}(n^2)$ storage, where n is the number of nucleotides in the sequence. This partition function calculation also provides (among other things) the base pairing probability for each possible base pair in the sequence, which can be displayed in a probability dot plot. It has been implemented in the Vienna RNA package [40, 39], which we use for calculation. The partition function algorithm could accommodate constraints by forcing base pairs or prohibiting base pairs. In the calculation of partition function, structures violating the constraints do not contribute to the partition function. The time and storage complexity are the same as that without constraints. Now let us compute the optimization in (3.4). In RNA secondary structure setting, the optimization becomes

$$\max_{(i,j)} I\left(A_{ij}; \left((i',j'), A_{i'j'}\right)\right), \tag{3.16}$$

where $I\left(A_{ij}; \left((i',j'), A_{i'j'}\right)\right) = \frac{1}{k-1}\sum_{(i',j')\neq(i,j)} I(A_{ij}, A_{i'j'})$, and k is the number of base pairs that differ from (i,j), i.e. $k = \frac{n^2-n}{2} - 1$. We first build a pairwise mutual information table, i.e. we compute the mutual information for a pair of different base pairs, i.e. $I(A_{ij}; A_{i'j'})$. We have

$$\begin{aligned} I(A_{ij}, A_{i'j'}) &= H(A_{ij}) - H(A_{ij}|A_{i'j'}) \\ &= H(A_{ij}) - \sum_{A_{i'j'}=0,1} P(A_{i'j'})H(A_{ij}|A_{i'j'}) \\ &= H(A_{ij}) - P(A_{i'j'}=1)H(A_{ij}|A_{i'j'}=1) - P(A_{i'j'}=0)H(A_{ij}|A_{i'j'}=0). \end{aligned} \tag{3.17}$$

As mentioned above, it takes $\mathrm{O}(n^3)$ to compute the marginal base pair probability $p_{ij} = P(A_{ij} = 1)$ [62] for the first term in (3.17). For the other two terms, we need the base pair probability conditioning on presence and absence of a certain base pair, which is another computation of time complexity $\mathrm{O}(n^3)$. Back to the optimization in (3.16), which needs to sum the pairwise mutual information for all possible base

pairs, thus the total time complexity required to compute the conditional entropy term for all pairs is O(n^3)*O(n^2)=O(n^5) (The number of possible pairs is O(n^2) and we need to compute the base pair probability conditioning on all different base pairs). In sum, although much reduced, the total time complexity for the optimization is O(n^5), which is still too expensive especially when the sequence length n is large. An alternative choice is empirical approach which we will discuss shortly.

Exact calculation of $H(A_{ij}|_{ij}A)$ We also provide here the exact calculation for the conditional entropy term which would be useful considered as constraints or as a standard to judge the power of the predicted informative base pair. We have

$$H(A_{ij}|_{ij}A) = H(A) - H(_{ij}A),$$

and we will discuss the computation for each term.

For an RNA molecule, the secondary structures in the Boltzmann ensemble are not all equally probable. The Boltzmann equilibrium probability of a specific secondary structure A for a sequence S is given by

$$P(A) = \frac{\exp\big(-E(A)/RT\big)}{U}, \tag{3.18}$$

where $E(A)$ is the free energy of the structure A for the sequence S, R is the gas constant, T is the absolute temperature and U is the partition function for all admissible secondary structures of the RNA sequence, i.e. $U = \sum_A \exp(-E(A)/RT)$. The Boltzmann equilibrium probability distribution gives the probability for every structure, and therefore statistically characterizes the ensemble. Here we would demonstrate the exact calculation using the simpler model, i.e. the stacking energy model as in Section 3.2.

- **Calculation of $H(A)$**

$$\begin{aligned} H(A) &= -\sum_A P(A)\log P(A) \\ &= -\sum_A \frac{\exp\big(-E(A)/RT\big)}{U}\big(-E(A)/RT - \log U\big) \\ &= \log U + \sum_A \frac{\exp\big(-E(A)/RT\big)E(A)/RT}{U} \end{aligned} \tag{3.19}$$

in (3.19), $\log U$ and U could be derived from partition function calculation and it remains to compute $\sum_A \frac{\exp\left(-E(A)/RT\right)E(A)}{RT}$. Based on the stacking energy model for base pair stacking, for a secondary structure A, its energy could be written as $E(A) = \sum_{(i,j)} EP(i,j)A_{ij}A_{i+1,j-1}$, where $EP(i,j) = E\big([i\cdot j/(i+1)]\cdot[(j-1)/RT]\big)$ is the stacking energy between the adjacent base pairs $r_i - r_j$ and $r_{i+1} - r_{j-1}$. Then, we have

$$\begin{aligned} &\sum_A \frac{\exp\Big(-E(A)/RT\Big)E(A)}{RT} \\ =&\sum_A \left(\prod_{(i,j)} \exp\Big(\frac{-EP(i,j)A_{ij}A_{i+1,j-1}}{RT}\Big)\right)\cdot\left(\sum_{(i',j')} \frac{EP(i',j')A_{i'j'}A_{i'+1,j'-1}}{RT}\right) \\ =&\sum_{(i',j')}\sum_A \left(\prod_{(i,j)} \exp\Big(\frac{-EP(i,j)A_{ij}A_{i+1,j-1}}{RT}\Big)\right)\cdot\left(\frac{EP(i',j')A_{i'j'}A_{i'+1,j'-1}}{RT}\right) \\ =&\sum_{(i',j')}\sum_A \left(\prod_{(i,j)\neq(i',j')} \exp\Big(\frac{-EP(i,j)A_{ij}A_{i+1,j-1}}{RT}\Big)\right)\cdot \\ &\left(\exp\Big(\frac{-EP(i',j')}{RT}\Big)\frac{EP(i',j')}{RT}A_{i'j'}A_{i'+1,j'-1}\right). \end{aligned} \tag{3.20}$$

We compute (3.20) by summing over all (i', j'), i.e. for each (i', j'), compute

$$\sum_A \left(\prod_{(i,j)\neq(i',j')} \exp\Big(\frac{-EP(i,j)A_{ij}A_{i+1,j-1}}{RT}\Big) \right) \cdot \left(\exp\Big(\frac{-EP(i',j')}{RT}\Big) \frac{EP(i',j')}{RT} A_{i'j'}A_{i'+1,j'-1} \right) \tag{3.21}$$

and then sum together. Compare (3.21) with the partition function

$$U = \sum_A \exp\Big(- E(S,A)/RT \Big) = \sum_A \prod_{(i,j)} \exp\Big(\frac{-EP(i,j)A_{ij}A_{i+1,j-1}}{RT}\Big),$$

the only difference is the term involving (i', j'), which is replaced by

$$\exp\Big(\frac{-EP(i',j')}{RT}\Big) \frac{EP(i',j')}{RT} A_{i'j'}A_{i'+1,j'-1}.$$

It could be interpreted as modifying the stacking energy for $r_{i'} - r_{j'}$ and $r_{i'+1} - r_{j'-1}$ to be $EP(i', j') - RT \log\Big(\frac{EP(i',j')A_{i'j'}A_{i'+1,j'-1}}{RT}\Big)$ and the stacking energy for all other base pairs remains unchanged. Then we can use the same algorithm used to calculate the partition function to compute (3.21). Let us now consider the time complexity. Although there is only one stacking energy term that differs from the original stacking energy model, we don't benefit much from the original dynamic programming result. The complexity is still $\mathrm{O}(n^3)$. We will illustrate the reasoning shortly. By summing over all possible base pairs, we are then able to compute the entropy $H(A)$ with total time complexity $\mathrm{O}(n^5)$.

- **Calculation of** $H({}_{ij}A)$ For a certain base pair (i, j), the method to compute

$H({}_{ij}A)$ is similar with $H(A)$ above.

$$P({}_{ij}A) = P({}_{ij}A, A_{ij} = 0) + P({}_{ij}A, A_{ij} = 1)$$
$$= \frac{\prod_{(i',j')\neq(i,j)} \exp\left(\frac{-EP(i',j')A_{i'j'}A_{i'+1,j'-1}}{RT}\right) \cdot \left(1 + \exp\left(\frac{-EP(i,j)A_{i+1,j-1}}{RT}\right)\right)}{U} \tag{3.22}$$

(3.22) has the same form as $P(A)$ and could be obtained from $P(A)$ by modifying the stacking energy term for $r_{i'} - r_{j'}$ and $r_{i'+1} - r_{j'-1}$ to be

$$-RT \log\left(1 + \exp\left(\frac{-EP(i,j)A_{i+1,j-1}}{RT}\right)\right).$$

The time complexity for a given pair is $\mathrm{O}(n^3)$ and the total time complexity required for optimization would be $\mathrm{O}(n^5)$.

- Can we save time since we only modify one stacking energy term?

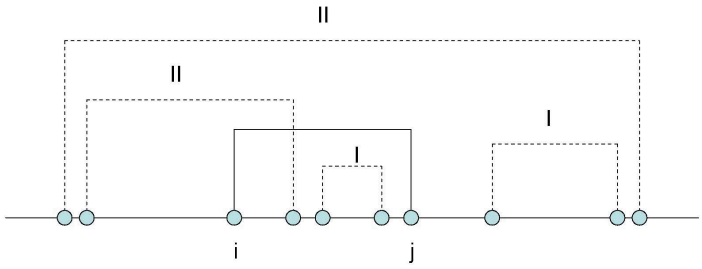


Figure 3.2: How many entries do we need to update?

Suppose now we only modify the stacking energy term for $r_i - r_j$ and $r_{i+1} - r_{j-1}$, then how many entries in the dynamic programming table could be reused and how much more calculation do we need compared to the original dynamic programming table for partition function? As demonstrated in Figure 3.2, for the grid corresponding to i', j' in dynamic programming table, if both i', j' satisfy $i \leq i', j' \leq j$ or $i', j' \leq i$ or $i', j' \geq j$ (type I in Figure 3.2), then this gird remains the same as the original table since the only stacking energy term

changed do not make any effect for the calculation of this grid. While for the i', j' otherwise (type II in Figure 3.2), the corresponding entry would be different from the original table. The terms we need to update has order $\mathrm{O}(n^2)$, which shows no much benefit from the original table since the total number of entries in the dynamic programming table is also $\mathrm{O}(n^2)$.

In summary, we provide an exact approach to calculate $H(A_{ij}|_{ij}A)$ and therefore also $I(A_{ij};_{ij} A)$. We fulfill this by modifying the stacking energy terms in the original dynamic programming algorithm and the overall time complexity is $\mathrm{O}(n^5)$.

Sampling based computation The previous paragraphs describe an algorithm with time complexity $\mathrm{O}(n^5)$ to implement the optimization in (3.16). This time complexity is too high especially for long sequences. As seen in Chapter 2, posterior samples are quite useful in inference and estimates. In this section we will provide an empirical calculation using samples drawn from the posterior space.

Ding and Lawrence extended the partition function calculation to compute a statistically valid sample of secondary structures in the Boltzmann ensemble and calculate sampling statistics of structural features [28, 30]. For every RNA sequence, we first cluster 1000 statistically sampled structures. The structure sample size of 1000 has been shown to be sufficiently large to guarantee statistical reproducibility in typical sampling statistics including base pair frequencies, even when two independent samples do not share a single structure [30]. In addition, regardless of sequence length, a cluster with an appreciable probability is expected to be represented in a sample of 1000 structures. Larger samples would reveal additional cluster at a significant level of 0.001. In this example we work with sample size $N = 1000$. Suppose that the N samples drawn from the posterior space of RNA secondary structure are $A^1, A^2, \cdots, A^N$.

- **Calculation of** $I(A_{ij}; A_{i'j'})$ In (3.17), the calculation of the first term $H(A_{ij})$ only requires the marginal pairwise probability p_{ij}. Let $X = 1_{\{A_{ij}=1, A_{i'j'}=1\}}$ and $Y = 1_{\{A_{i'j'}=1\}}$ be the indication function of corresponding conditions. From

the N samples we have corresponding $X_1, X_2, \cdots, X_N$ and $Y_1, Y_2, \cdots, Y_N$. Applying Law of Large Number,

$$\lim_{k\to\infty} \bar{X}_k = EX = P(A_{ij} = 1, A_{i'j'} = 1)$$

$$\lim_{k\to\infty} \bar{Y}_k = EY = P(A_{i'j'} = 1)$$

Then,

$$\lim_{k\to\infty} \frac{\bar{X}_k}{\bar{Y}_k} = \frac{EX}{EY} = \frac{P(A_{ij} = 1, A_{i'j'} = 1)}{P(A_{i'j'} = 1)} = P(A_{ij} = 1 | A_{i'j'} = 1).$$

Therefore we could estimate the marginal pairwise probability under constraint $A_{ij} = 1$ (or $A_{ij} = 0$) by

$$\hat{P}(A_{ij} = 1 | A_{i'j'} = 1) = \frac{\#\{A^k \text{ with } A_{ij}=1 \text{and} A_{i'j'}=1\}}{\#\{A^k \text{ with } A_{i'j'}=1\}}.$$

Denote $ent(p) = p\log(\frac{1}{p}) + (1-p)\log(\frac{1}{1-p})$, then the estimator for $I(A_{ij}, A_{i'j'})$ could be

$$\begin{aligned} I(A_{ij}, A_{i'j'}) =& H(A_{ij}) - P(A_{i'j'} = 1) ent\Big(\hat{P}(A_{ij} = 1 | A_{i'j'} = 1)\Big) \\ & - P(A_{i'j'} = 0) ent\Big(\hat{P}(A_{ij} = 1 | A_{i'j'} = 0)\Big), \end{aligned} \tag{3.23}$$

which converges in probability according to Law of Large Number. Using Chebyshev inequality, we have

$$P(|\bar{X}_N - E\bar{X}_N| > \epsilon) \leq \frac{Var(\bar{X}_N)}{\epsilon^2}$$

since

$$E\bar{X}_N = P(A_{ij} = 1, A_{i'j'} = 1)$$

$$Var(\bar{X}_N) = \frac{1}{N} P(A_{ij} = 1, A_{i'j'} = 1)\Big(1 - P(A_{ij} = 1, A_{i'j'} = 1)\Big),$$

then,

$$\begin{aligned}
&P(|\bar{X}_N - P(A_{ij}=1, A_{i'j'}=1)| \geq \epsilon) \\
\leq &\frac{1}{N\epsilon^2} P(A_{ij}=1, A_{i'j'}=1)\Big(1 - P(A_{ij}=1, A_{i'j'}=1)\Big)
\end{aligned}$$

Hence, $\hat{P}(A_{ij}=1, A_{i'j'}=1)$ converges to $P(A_{ij}=1, A_{i'j'}=1)$ in probability, and same for other similar terms. Also, from Central Limit Theorem, we would have $\frac{\sqrt{N}(\bar{X}_N - E(X))}{\sqrt{Var(X)}} \sim N(0,1)$, then the L_2 norm $\parallel \bar{X}_n - E(X) \parallel_2$ has convergence order $O(\frac{1}{\sqrt{n}})$. Since $I(A_{ij}, A_{i'j'})$ is a continuous function of such terms, thus, we have $I(A_{ij}, A_{i'j'})$ also converges in probability.

- **Calculation of** $H(A_{ij}|_{ij}A)$

$$\begin{aligned}
H(A_{ij}|_{ij}A) &= -\sum_{{}_{ij}A} \sum_{A_{ij}} P(A_{ij},{}_{ij}A) \log P(A_{ij}|_{ij}A) \\
&= -\sum_{{}_{ij}A} \sum_{A_{ij}} P(A) \log \frac{P(A)}{P({}_{ij}A)} \\
&= -\sum_{{}_{ij}A} \sum_{A_{ij}} P(A) \log \frac{P(A)}{\Big(P({}_{ij}A, A_{ij}=0) + P({}_{ij}A, A_{ij}=1)\Big)} \\
&= -E\left(\log \frac{P(A)}{\Big(P({}_{ij}A, A_{ij}=0) + P({}_{ij}A, A_{ij}=1)\Big)}\right) \qquad (3.24)
\end{aligned}$$

According to Law of Large Number, we could estimate the expectation in (3.24) by Monte Carlo method, i.e. sample mean. By Monte Carlo method, it requires the probabilities for each sample. Now consider the cases in which we are unable to obtain the normalization constant, e.g. when using an MCMC sampler, when write out the expressions we could see that the normalization constant term is actually cancelled out. Note that in the energy model we discuss here, we could easily get the probability for each given structure, by dividing corresponding energy by the partition function U. The estimator

could be written as below and it converges in probability.

$$\begin{aligned} & E\left(\log \frac{P(A)}{\Big(P({}_{ij}A, A_{ij}=0)+P({}_{ij}A, A_{ij}=1)\Big)}\right) \\ \approx & \frac{1}{N}\sum_s \log \frac{P(A^s)}{P\Big({}_{ij}A^s, A^s_{ij}=0)+P({}_{ij}A^s, A^s_{ij}=1\Big)} \end{aligned} \tag{3.25}$$

Note that $P(A^s) = \frac{\exp\Big(-E(A^s, I_{1n})/RT\Big)}{U}$, where U is the partition function. Then

$$\begin{aligned} & \frac{P(A^s)}{P({}_{ij}A^s, A^s_{ij}=0)+P({}_{ij}A^s, A^s_{ij}=1)} \\ = & \frac{\exp\Big(-E(A^s)/RT\Big)}{\exp\Big(-E({}_{ij}A^s, A^s_{ij}=0)/RT\Big)+\exp\Big(-E({}_{ij}A^s, A^s_{ij}=1)/RT\Big)}, \end{aligned} \tag{3.26}$$

which does not require the calculation of partition function U. Thus, our sampling approach is also applicable to MCMC samplings, which draw samples in proportion to their probabilities without calculating total partition function.

Let us consider again the estimation error. Let X denote the random variable $\log \frac{P(A)}{\Big(P({}_{ij}A, A_{ij}=0)+P({}_{ij}A, A_{ij}=1)\Big)}$, and $X_1, \cdots, X_n$ to denote the corresponding sample values. Then the L.H.S of (3.25) is $E(X)$, and the R.H.S is $\bar{X}_N$. From Chebyshev inequality,

$$P(|\bar{X}_N - E(\bar{X}_N)| > \epsilon) = P(|\bar{X}_N - E(X)| > \epsilon) \leq \frac{Var(\bar{X}_N)}{\epsilon^2} = \frac{Var(X)}{N\epsilon^2}.$$

Furthermore, from Central Limit Theorem, we have $\frac{\sqrt{N}(\bar{X}_N - E(X))}{\sigma} \sim N(0,1)$, where $\sigma = \sqrt{Var(X)}$. Therefore, The L_2 norm $\|\bar{X}_N - E(X)\|_2$ has convergence order $O(\frac{1}{\sqrt{N}})$.

- **Calculation of $H(A)$** Calculation of the total entropy is important since it could be used as standard or criteria when we judge the robustness of the

informative variables we found.

$$H(A) = -\sum P(A) \log P(A) = E \log \frac{1}{P(A)}$$

We could use Monte Carlo method to estimate $H(A)$ by samples, in which we use empirical frequency as an estimate for $P(A)$. However, with same argument in previous paragraph, we notice that we could easily get the probability for each given structure. Therefore, we tend to use the following estimate:

$$H(A) = \frac{1}{N} \log \Big(\frac{1}{P(A^s)} \Big). \tag{3.27}$$

Recall the empirical calculation of $I(A_{ij}; A_{i'j'})$, we use empirical probability (also known as relative frequency) to estimate $P(A_{ij} = 1, A_{i'j'} = 1)$ and $P(A_{i'j'} = 1)$. Empirical probability is the ratio of the number of "favorable" observations to the total number of observations. From another point of view, empirical probability could be considered as the sample mean of the random variable with value one for occurrence and zero otherwise. The expectation of this random variable is the true probability and thus empirical probability is an unbiased estimator for probability from Law of Large Number. This is actually widely used as Monte Carlo integration, which has convergence rate of $\frac{1}{\sqrt{N}}$. Note that here Monte Carlo Integration only involves observations in order to calculate. However, in many cases the samples are drawn in proportion to their true probability (or density for continuous case), e.g. Monte Carlo Sampling. Therefore, in these circumstance not only do we have observations, we also have corresponding probability or subject to a normalizing constant. Come back to empirical calculation: compared with empirical probability, (3.25) (3.27) is also a Monte Carlo Integration approach, however, the integration function is a function of some probability terms of observation which could be easily computed given observation. Therefore, instead of using empirical probability as estimator for these terms, we use the true probability computed from observation on the R.H.S of (3.25) (3.27). The true probability is different from the empirical probability, especially

when the space is huge. For example, in a high dimension space, if the number of ensemble is huge while the probability for each individual is small, then it is very likely that each observation only occur once in the N samples drawn. Then the empirical probability is $\frac{1}{N}$ for all observations. A straightforward thinking is that shall we give weight to the observations in accordance to the proportion of their probability? We will discuss it in Chapter 4. Let us go back to (3.25) (3.27) . Now that we make a small modification to the estimator by using the true probability instead of the empirical probability, although at the first glance we use more information in addition to the drawn samples, we still need to ask whether it is a better estimator than the standard empirical one using only samples?

In more mathematical terms: consider a discrete high dimensional random variable X having probability mass function $p(x)$ which is greater than zero on a set of values $\mathcal{X} = \{x_1, x_2, \cdots, x_m\}$. Then the expected value of a function G of X is

$$E[G(X)] = \sum_{x \in \mathcal{X}} G(x)p(x).$$

If we were to take an N-sample of X's, $(X_1, X_2, \cdots, , X_N)$, then we would have the Monte Carlo estimate

$$\tilde{G}_N(X_1^N) = \frac{1}{N} \sum_{i=1}^{N} G(X_i). \tag{3.28}$$

Now, if $G(x) = J(p(x))$ is a function of the mass function, which term shall we use for $G(X_i)$ in (3.28)? If we use samples only and use the empirical frequency as the estimator for $p(X_i)$, we have

$$\tilde{G} = \frac{1}{N} \sum_{i=1}^{N} J(\tilde{p}(X_i)) = \frac{1}{N} \sum_{i=1}^{N} J(\tilde{p}_i), \tag{3.29}$$

where $\tilde{p}_i = \frac{1}{N} \sum_{j=1}^{N} 1_{\{X_j = X_i\}}$ is the empirical frequency for X_i. If in addition to the samples, we also have the corresponding probabilities $p_1, p_2, \cdots, p_N$, we might have another estimator

$$\hat{G} = \frac{1}{N} \sum_{i=1}^{N} J(p(X_i)) = \frac{1}{N} \sum_{i=1}^{N} J(p_i), \tag{3.30}$$

which uses the exact probability p_i.

To investigate and compare the behaviors of these two estimators, we have the following two Lemmas.

Lemma 3.2.1. *If $J(.)$ is a continuous function, then $\tilde{G}$ is asymptotically unbiased estimator for $E[G(X)]$, and $\hat{G}$ is unbiased estimator.*

Proof. When N tends to infinity, we have

$$\begin{aligned}
\lim_{N\to\infty} E[\tilde{G}] &= E[\lim_{N\to\infty} \tilde{G}] \\
&= E[\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} J(\tilde{p}(X_i))] \\
&= E[\lim_{N\to\infty} \sum_{i=1}^{m} \tilde{p}(x_i)J(\tilde{p}(x_i))] \\
&= E[\sum_{i=1}^{m} p(x_i)J(p(x_i))] \\
&= E[E[G(X)]] \\
&= E[G(X)].
\end{aligned}$$

Since $J(.)$ is a continuous function and $\tilde{p}(x)$ is bounded, then $\tilde{G}(x) = J(\tilde{p}(x))$ is also bounded. Therefore, we could exchange the order of limitation and expectation in the first equivalence. The first and the third equivalences are straightforward. The fourth equivalence is according to the Law of Large Number.

The proof for unbiased property of $\hat{G}$ could be easily derived.

$$\begin{aligned}
E[\hat{G}] &= \frac{1}{N}\sum_{i=1}^{N} E[J(p(X_i))] \\
&= \frac{1}{N} \cdot N \cdot E[J(p(X))] \\
&= E[J(p(X))] \\
&= E[G(X)]
\end{aligned}$$

□

Lemma 3.2.2. *If $J(.)$ is a continuous function, then both $\tilde{G}$ and $\hat{G}$ converge to $E[G(X)]$ when N tends to infinity.*

Proof. The proof for $\tilde{G}$ is essentially the same as in the previous lemma.

$$\begin{aligned}\lim_{N\to\infty} \tilde{G} &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} J(\tilde{p}(X_i)) \\ &= \lim_{N\to\infty} \sum_{i=1}^{m} \tilde{p}(x_i) J(\tilde{p}(x_i)) \\ &= \sum_{i=1}^{m} p(x_i) J(p(x_i)) \\ &= E[G(X)]\end{aligned}$$

$$\begin{aligned}\lim_{N\to\infty} \hat{G} &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} J(p(X_i)) \\ &= \lim_{N\to\infty} \sum_{i=1}^{m} \tilde{p}(x_i) J(p(x_i)) \\ &= \sum_{i=1}^{m} p(x_i) J(p(x_i)) \\ &= E[G(X)]\end{aligned}$$

□

In Lemma 3.2.1, we prove that $\tilde{G}$ are asymptotically unbiased and $\hat{G}$ is unbiased. In Lemma 3.2.2 we also show both $\tilde{G}$ and $\hat{G}$ converge to $E[G(X)]$. Then what else could we tell regarding which is better and which ought to be used? The following theorem gives some circumstance under which $\hat{G}$ have at least half probability to behave better than $\tilde{G}$.

Theorem 3.2.3. *Suppose the ensemble of a discrete high-dimensional space is large and satisfies the condition that for every point value x_i, $P(x_i) << \frac{1}{N}$ for the sample size N. We also assume that the N samples are different from each other, i.e. each*

with empirical frequency $\frac{1}{N}$*. Then when* $J(.)$ *is a monotone continuous function,*

$$P(|\tilde{G} - E[G(X)]| \geq |\hat{G} - E[G(X)]|) \geq \frac{1}{2} \tag{3.31}$$

Proof. Without loss of generality, assume that $J(.)$ is a monotonic decreasing function.

First, notice that under the condition that the empirical frequencies for all points appearing in the N samples are $\frac{1}{N}$, we have

$$\tilde{G} = \frac{1}{N}\sum_{i=1}^{N} J(\tilde{p}_i) = J(\frac{1}{N}). \tag{3.32}$$

Since $P(x) << \frac{1}{N}$ for all x,

$$\begin{aligned} E[G(X)] &= \sum_{i=1}^{m} p(x_i)J(p(x_i)) \\ &> \sum_{i=1}^{m} p(x_i)J(\frac{1}{N}) = J(\frac{1}{N}). \end{aligned} \tag{3.33}$$

$$\begin{aligned} \hat{G} &= \frac{1}{N}\sum_{i=1}^{N} J(p_i) \\ &> \frac{1}{N}\sum_{i=1}^{N} J(\frac{1}{N}) = J(\frac{1}{N}). \end{aligned} \tag{3.34}$$

Given (3.32)(3.33)(3.34) above, since

$$|\tilde{G} - E[G(X)]| = E[G(X)] - \tilde{G} = E[G(X)] - J(\frac{1}{N}),$$

then we have

$$|\hat{G} - E[G(X)]| \leq |\tilde{G} - E[G(X)]| \Leftrightarrow \hat{G} \leq 2E[G(X)] - J(\frac{1}{N}). \tag{3.35}$$

Let $d = 2E[G(X)] - J(\frac{1}{N})$ and $\delta = P(\hat{G} \leq 2E[G(X)] - J(\frac{1}{N}))$, then

$$\begin{aligned} E[G(X)] &= E[\hat{G}] \\ &\geq (1-\delta)d + \delta J(\frac{1}{N}) \\ &= (1-\delta)(2E[G(X)] - J(\frac{1}{N})) + \delta J(\frac{1}{N}). \end{aligned} \tag{3.36}$$

Solve (3.36) for δ, we get

$$\begin{aligned} \delta &= P(\hat{G} \leq 2E[G(X)] - J(\frac{1}{N})) \\ &\geq \frac{2(E[G(X)] - J(\frac{1}{N}))}{E[G(X)] - J(\frac{1}{N})} \\ &= \frac{1}{2}. \end{aligned} \tag{3.37}$$

Therefore from (3.35), we finish the proof. □

Theorem 3.2.3 would be applicable to the estimation of expectation where the function taken is a monotone function of $p(x)$, e.g. entropy or mutual information etc, since

$$H(X) = E\big[\frac{1}{P(X)}\big],$$

and $J(x) = \frac{1}{x}$ is monotone.

In summary, in this section we provide both exact and empirical approaches to calculate the pairwise mutual information and conditional entropy terms. Besides, we prove that under some circumstance, using "correct" probability terms in the Monte Carlo estimation of the expectation of some function of $p(x)$ would not only give an unbiased estimator, furthermore this estimator would have at least half probability to have a better performance than the usual estimation using only the samples.

3.2.3 Examples

We select several RNA sequences from different structural RNA classes from online RNA databases (http://www.ncbi.nlm.nih.gov/nuccore). For each sequence, 1000 structures are sampled by Vienna Package [39] and clustered by *Diana* approach

[48] and Sfold [17]. Although the structural RNA may have unique structure in solution, Ding found that there usually exists a small number of distinct clusters of similar structures in the Boltzmann weighted ensemble. Although the number of conformational structures grows exponentially with sequence length, we found no evidence that the number of clusters would increase with the length of the nucleotide sequence. Now we show some results for several RNA sequences.

Example 1: *Agrobacterium tumefaciens* 5S rRNA, GenBank: X02672.1

Hierarchical Clustering of 5S RNA Sfold samples structures in accordance with their Boltzmann weighted probabilities, the probability of a cluster is estimated by the frequency of structures in that cluster, i.e., the number of structures in the cluster divided by the sample size. In the case of multiple occurrences of the same structure, each occurrence is counted in the calculation. The probabilities of the two clusters are 0.631 and 0.369 respectively. Figure 3.3 plots the circle diagram of these two cluster centroids [14] and Figure 3.4 gives the Minimum Free Energy (MFE) structure. In this example he MFE, the most probable structure falls in the smaller cluster. In Figure 3.3, the structures of these two centroid are very different from each other. They share a few common stems, and have some particular stems of their own. And it is often the case that taken together their characteristic stems form a pseudoknot. In this 5S rRNA, notice that the first cluster centroid could be characterized by stems formed by base pairs around $39-100$ and the second cluster centroid could be characterized by stems formed by base pairs around $21-64$.

Pairwise Mutual Information Optimization for 5S RNA Maximizing (3.16) with constraint (3.6) and (3.7) by our estimators in the previous section gives the most information base pair $21-64$, whose marginal base pair probability is 0.3553 and averaged pairwise mutual information is 0.0533. Base pair $21-64$ belongs to the stems of the centroid clustering result from Sfold. Good match! Figure 3.5 gives the histogram and EDF plot of individual terms in the averaged mutual information for 21-64. As we could see, there are a large amount of base pairs have very low mutual

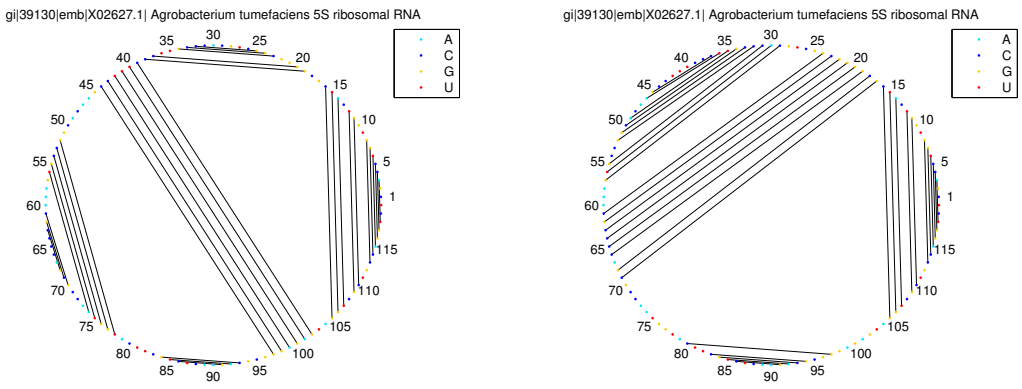


Figure 3.3: **Cluster Centroid of 5S rRNA.** Circle diagrams of the cluster centroids for two distinct clusters (arranged in descending order of cluster size), for the 5S rRNA sequence of length 120nt. In a circle diagram, bases are positioned along a circle in the clockwise orientation, and a base-pair is shown by an arc that connects the two bases.

information with respect to base pair $21-64$. Actually there are several categories that give low mutual information: (1) base pairs with marginal probabilities close to 0 (thus with low entropy), since mutual information is always no more than the entropy of either variable. (2) base pairs with marginal probabilities close to 1 (thus with low entropy). (3) other base pairs that have high dependency with base pair $21-64$, e.g. the neighbor base pairs or crossing base pairs from the stems of the other characteristic cluster.

Figure 3.6 gives the centroid in the subspaces with and without the base pair $21-64$. Compared with the centroid structures of the two clusters from Sfold in Figure 3.3, the centroid structures of each cluster are very similar to those from Sfold and the base pair we found do represent each cluster. Next, we partition the posterior space into two disjoint subspaces by presence or absence of this most informative base pair. Now conditioned on presence of base pair $21-64$, the most informative base pair we found is $24-54$ with averaged mutual information 0.0093. Conditioned on absence of this base pair, the most informative base pair we found is $54-76$ with average mutual information 0.0119. Figure 3.7 and Figure 3.8 give the centroids of the clusters in corresponding subspaces.

The advantage of using informative base pairs to characterize the posterior space is that we are able to describe the distribution over each subspace and draw samples

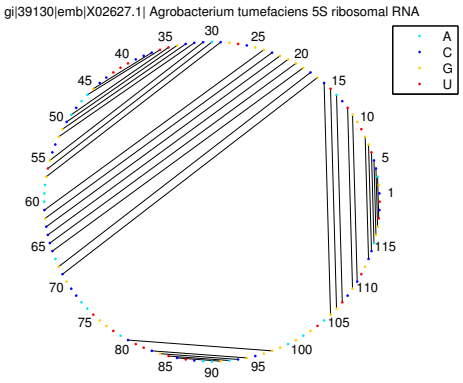


Figure 3.4: **MFE structure of 5S rRNA.** The MFE structure is present in the larger cluster.

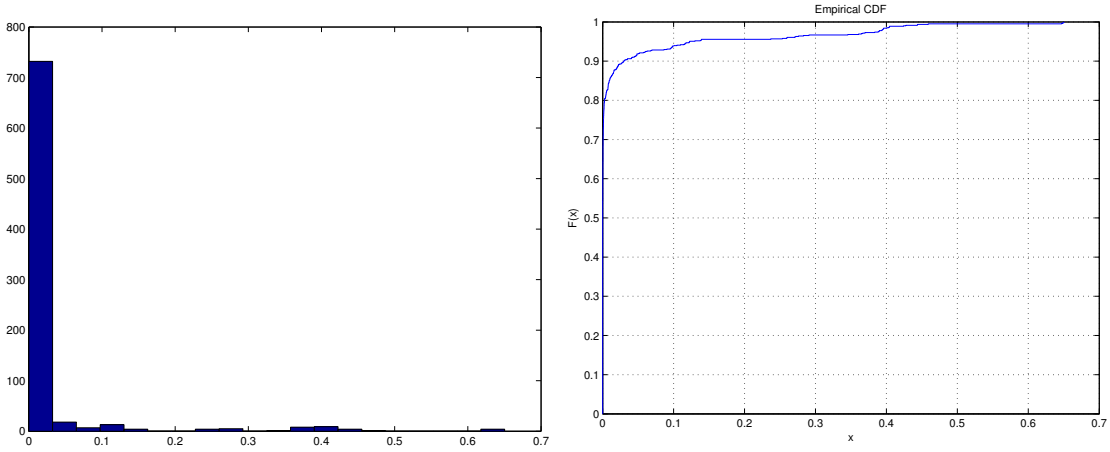


Figure 3.5: Histogram and EDF of individual term in the averaged mutual information for 21-64.

from each subspace. In addition, when the RNA is just of stable structural class, as expected for structure RNAs, knowing about the presence or absence of the most informative base pair can greatly improve structural prediction. As we look at the ordered averaged pairwise mutual information list, we would find some helpful information. Following the most informative base pair $21-64$, the next most informative base pairs are $22-63$, $20-65$, $23-62$ etc, i.e. those forms the stem together with $21-64$. After these, base pairs $42-99$, $41-100$, $43-98$ follows. We could see that these base pairs are from the important stem of the other cluster. Traversing the sorted averaged pairwise mutual information list, we are able to discover the

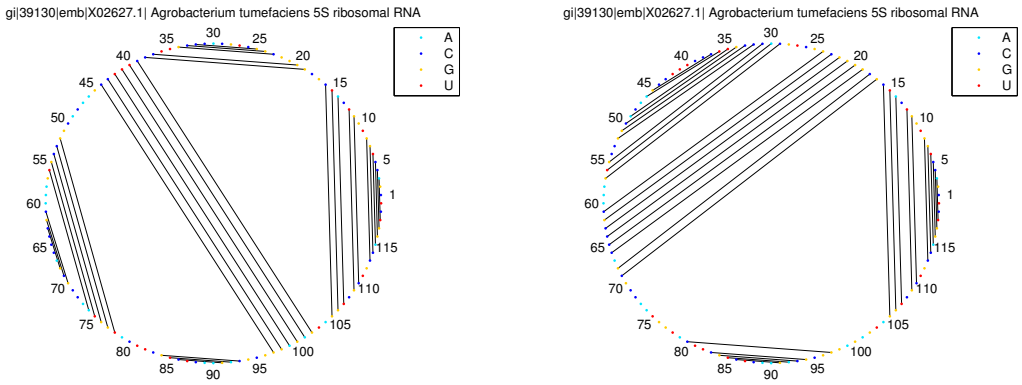


Figure 3.6: Cluster Centroid of 5S rRNA (with and without base pair 21-64.

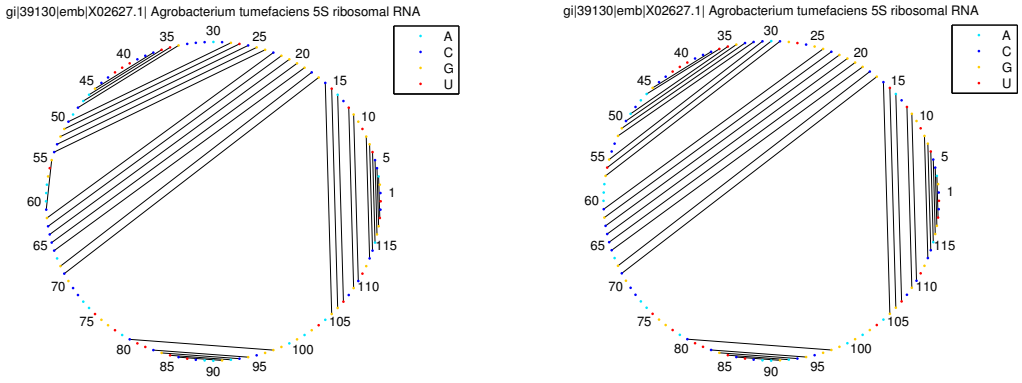


Figure 3.7: **Cluster Centroid in the subspace with existence of base pair** $21-64$**.** Within this subspace, the most informative base pair is $24-54$, and these are the corresponding centroid in the sub-subspace of existence or absence of $24-54$. We could observe different stems in that area.

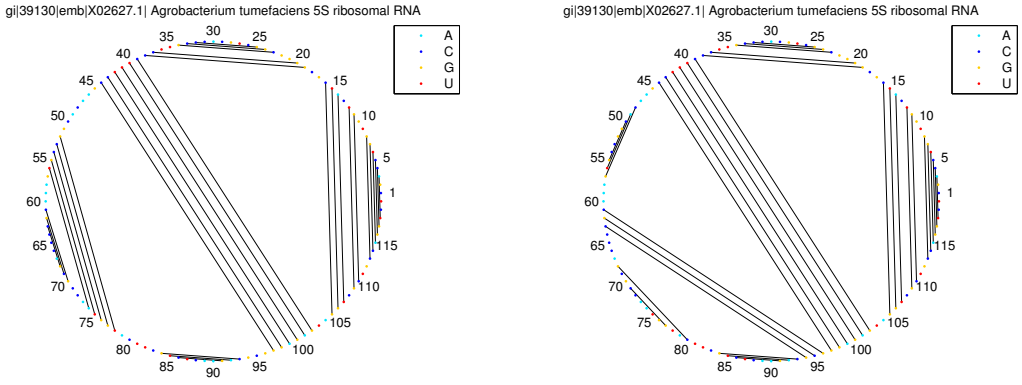


Figure 3.8: **Cluster Centroid in the subspace with absence of base pair** $21-64$**.** Within this subspace, the most informative base pair is $54-76$, and these are the corresponding centroid in the sub-subspace of existence or absence of $54-76$. We could observe different stems in that area.

important stems in the underlying secondary structures.

Example 2: Leptospirillum ferrooxidans strain CF1 RNAase P (rnpB) gene, GenBank: AF296042.1

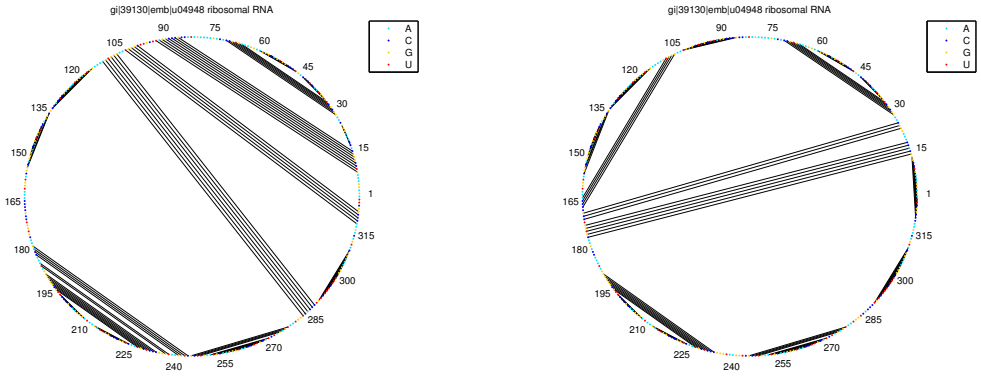


Figure 3.9: **Cluster Centroid of AF296042.** Circle diagrams of the cluster centroids for two distinct clusters (arranged in descending order of cluster size), for the AF RNA sequence of length 327nt. In a circle diagram, bases are positioned along a circle in the clockwise orientation, and a base-pair is shown by an arc that connects the two bases.

The most informative base pair we found is $107 - 287$ with averaged pairwise mutual information 0.0884 and marginal probability 0.8870. The first several most informative base pairs are the base pairs adjacent to base pair $107-287$, i.e, $106-288$, $108 - 288$ etc. These base pairs are just from the stem of the centroid of the first cluster. Following these base pairs, the next most informative base pairs are $11-90$, $10-91$ etc with averaged pairwise mutual information about 0.0764, then $17-174$, $16-175$ etc with averaged mutual information about $0.0689\cdots$. Conditioned on presence of the most informative base pair $107-287$, the next most informative base pair is $156-281$ with average mutual information 0.0548 and marginal probability 0.1321. For the subspace without $107-287$ base pair, the mass proportion is only 0.1130 and we are not going to divide it further.

Example 3: Homo sapiens dystrophin (DMD), transcript variant Dp40, mRNA, GenBank: NM004019

In the clustering result from Sfold, the mass proportion of cluster one and cluster

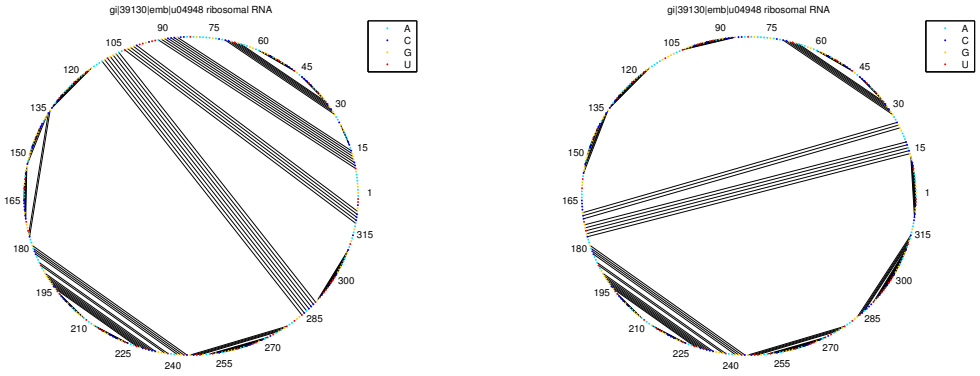


Figure 3.10: Cluster Centroid of AF296042 with and without base pair $107 - 287$

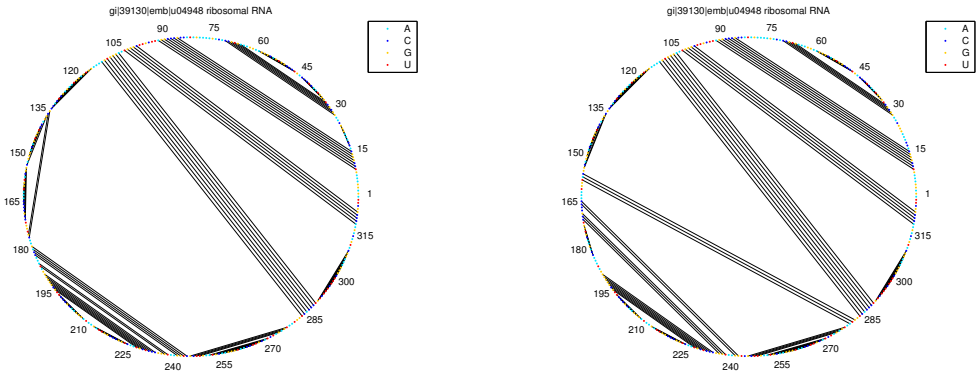


Figure 3.11: Cluster Centroid of AF296042 in presence of base pair $107 - 287$, with and without $156 - 281$

two are 0.891 and 0.109 respectively. The most informative base pair we found is $570-768$ with averaged mutual information 0.0973 and marginal probability 0.8428. The next few most informative base pairs are the base pairs adjacent to base pair $570-768$. This is just the stems of the centroid of the first cluster. After these base pairs, the next most informative base pairs we found are those around $1190-1388$, then along with base pairs around $84-388$ and $70-538$. Conditioned on presence of the most informative base pair $570-768$, the next most informative base pair our algorithm found is $84-388$ with averaged mutual information 0.0475 and marginal probability 0.3643.

This is an example in which almost all ensembles have similar structures. And our algorithm just try its best to look for the informative base pairs.

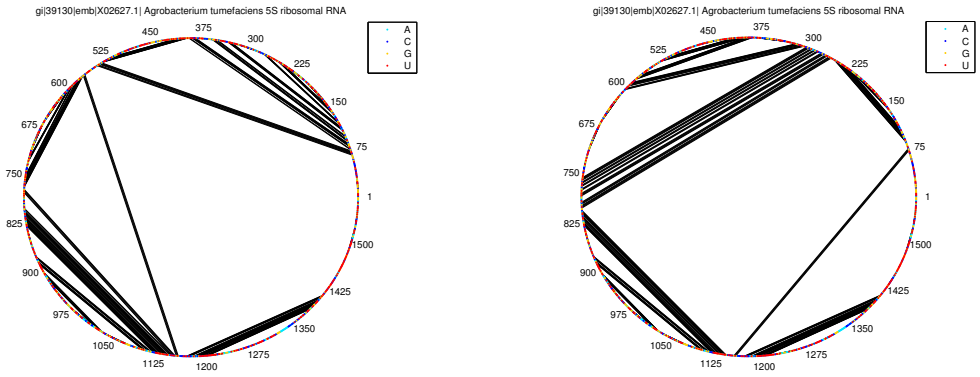


Figure 3.12: **Cluster Centroid of NM004019.** Circle diagrams of the cluster centroids for two distinct clusters (arranged in descending order of cluster size), for the AF RNA sequence of length 1571nt. In a circle diagram, bases are positioned along a circle in the clockwise orientation, and a base-pair is shown by an arc that connects the two bases.

In summary, from the three examples above, we could see that our algorithm is able to locate the important stems in the RNA secondary structure. Specifically, the base pairs with high averaged pairwise mutual information are usually the important and characteristic stems in the centroid structure of each cluster. In these examples, we obtain similar clusters with those from Sfold, while our algorithm is able to characterize each cluster by only one or two characteristic base pairs.

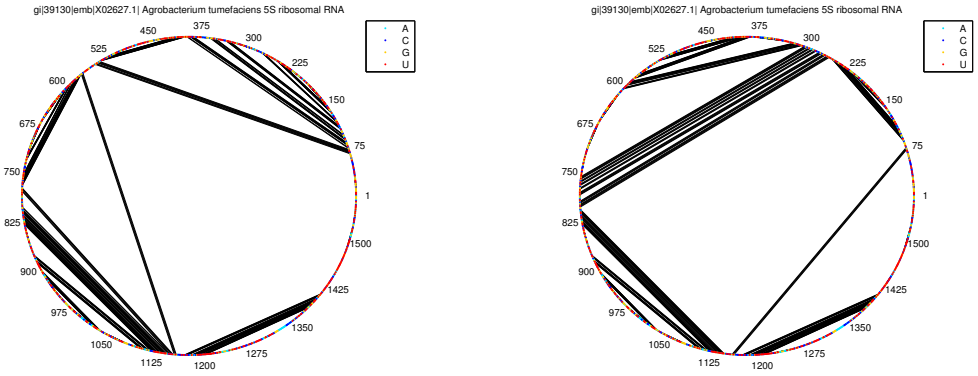


Figure 3.13: Cluster Centroid of NM004019 with and without base pair $570 - 768$

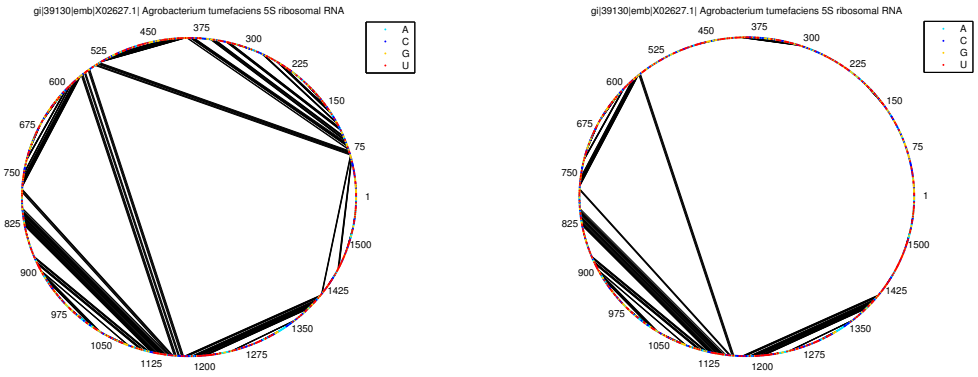


Figure 3.14: Cluster Centroid of NM004019 in presence of base pair $570 - 768$, with and without $84 - 388$

3.2.4 Discussion

Due to the ability of RNA to encode and manipulate genetic information, RNA is the most versatile biological macromolecule. RNA have flexible ways to fold on itself to form various biologically functional structures. Comparative sequence analysis is the most reliable way to establish secondary structure for RNA. However, this approach is not always applicable, it is only good for a small fraction of RNAs that have a large number of diverse and alignable, homologous structures. As mentioned in the introduction, minimum free energy [61, 76] is another approach to predict RNA secondary structure. As pointed out in [32], a third general approach to probe the secondary structure is experimental. However, in order to get a deep insight into the folding structure of RNA, the only possibility is to test the reactivity of every nucleotide. To fulfill this, enzymatic probing and lead-induced cleavages have been developed. It is possible to get the information of the pairing state of each nucleotide using a combination of probes with different specificities. In-line probing has been proven useful for developing secondary structure models for small RNAs and for quantifying ligand-induced conformational changes in RNA [71]. SHAPE chemistry [83] interrogates local nucleotide dynamics in RNAs of virtually any size in a single experiment. In addition to chemical probes, several RNases have been shown to react with single-stranded or unstructured regions in RNA, independent of the underlying sequence. These sequence-independent chemistries and RNase reagents make it possible to infer the state for every nucleotide base in the underlying RNA sequence.

With the information from chemical mapping and microarray experiments, the accuracy of computational methods to predict secondary structure might be improved by providing constraints that identify unpaired nucleotides. [36] demonstrates that incorporating the information of unpaired nucleotides from simple nuclear magnetic resonance (NMR) experiments as constraints into a dynamic folding algorithm greatly reduces the ensemble space and therefore increases the accuracy of prediction. NMR could also be able to produce complementary experimental constraints

that identify base pairs.

As the clusters identified by the most informative base often have large change in which bases are paired differently, combining the most informative base pair clusters with experimental data from these methods often promise to greatly improve structure prediction. Most informative base pairs might be quite helpful in the prediction of RNA secondary structure. As mentioned above, RNA secondary structure might be characterized by a set of stacking base pairs and most informative base pairs contain the information of these "important" stackings. Therefore, if we are able to detect the existence of these base pairings, we might get a good insight into the prediction of the secondary structure. Due to the small number of informative base pairs needed to describe the centroid structure, we only need a small number of probing to make our prediction. Furthermore, for RNAs that have a large number of diverse homologs, the fraction of being paired for a certain nucleotide base might also reveal important information by comparing this proportion with the marginal pairing probabilities in different subspaces.

3.3 Conclusion

In this chapter, we develop a characterization criteria based on information measure and design an iterative algorithm to identify the most informative variables in high dimensional spaces in order to describe the shape of the underlying probability spaces. We applied this approach to the posterior space of RNA secondary structure and employ the most informative base pairs to cluster the space.

Like feature selection, our most informative variable approach might not work well for all probability spaces, especially when the space is particular complex, e.g. totally uniform over all variables. For spaces with exclusivity constraints, e.g. the non-pseudoknot constraint for RNA secondary structures, we believe the most informative variables would be more likely to help describe the shape of the underlying probability spaces since the exclusivity constraint would divide the space into exclusive subspaces and the variables involved would be related to each other to a large

degree.

Furthermore, from the examples illustrated in Section 3.2.3, we notice that there actually exist a large number of base pairs with marginal probability close to 0 or 1, which thereby have low amount of entropy, i.e these base pairs contain quite small amount of information and are non-informative. This motivates us to use a reduced model, in which we put constraints for those base pairs with close to zero entropy, i.e. put the constraint that base pairs are absent for those with marginal probability close to zero and the constraint that base pairs are present for those with marginal probability close to one. This reduced model should reveal similar behavior with the complete model, while the reduced model reduced the dimensionality of the space to a large degree. Also, with the reduced model, we get rid of those non-informative base pairs, thus the averaged pairwise mutual information obtained from the optimization should give better idea of the amount of information shared between variables.

In addition, it seems putting different weight on the pairwise entropy term might make more sense compared to equal weight for all terms. We might also explore toward this direction to improve upon the current method.

Chapter 4

Characterize high dimensional data with their Reveal distributions: with application to clustering

There are huge number of probability models of applications in mathematics, physics, chemistry etc, in which not only can we draw samples following their probability distributions but also we are able to calculate corresponding probabilities or densities (maybe up to a normalization constant) of these samples, for example: Ising model, hidden Markov model, compound distribution, mixture models and so on. The posterior space of RNA secondary structure as illustrated in Chapter 3 is also such a situation which motivated our thinking of this particular problem. In Chapter 3, the application of our algorithm to the posterior space of RNA secondary structure desires to calculate the entropy, conditional entropy and mutual information of underlying random variables, which could actually be written as the expectation of some functions involving density or probability function $p(x)$. The mass function might be small for all ensembles in high dimensional spaces, which would lead to the fact that each ensemble might only appear once in the drawn N samples. Therefore,

if we go completely empirically and employ Monte Carlo method, we would encounter the problem that the empirical probability becomes $\frac{1}{N}$ for all the samples where N is the number of samples drawn. Theorem **3.2.2** proposes a modified Monte Carlo estimator to approximate the expectation, which replaces the empirical probability with the true probability or density in the standard Monte Carlo estimator and we prove that with at least half probability this new estimator would be closer to the true value than the standard Monte Carlo estimator.

Again, under these circumstance we have not only independent samples "x" drawn from the underlying distribution, but also along with their corresponding probabilities or densities "$p(x)$" and we would rather name it "x-px" problem after this consideration. Many questions arise given both "x" and "$p(x)$". For example, in high dimensional spaces, we are interested in characterizing the distribution of probability models. We would like to know how the distribution looks like, how many big modes exist in the density function and how many substantial clusters are there when we sample from the distribution. Of course, if we have samples, we can easily get clusters using some traditional clustering algorithms, like K-means, nearest neighbor method or some parameter approaches, however, can we obtain better performance if we also have probabilities or densities of samples? Also, K-means and nearest neighbor method do not reveal the shape of the underlying density function.

For distributions on one-dimensional or two-dimensional Euclidean spaces, we are able to plot the density functions in order to find out their characteristics, e.g. how many big modes or clusters. However, in high dimensional space, clustering or other characteristics of distributions deserve more investigation, since there is no picture to illustrate us. There are already many clustering methods in literature, for example K-means clustering, model-based clustering, e.g. Mixture of Gaussian distributions and so on. K-means [57] is one of the simplest unsupervised learning algorithms. K-means follows an easy and simple approach to classify a given set of data by a certain number of clusters. The idea behind K-means is to define different centroid to each cluster in a cunning way. In order to get a better shape of the samples, it is better to place them far away from each other as much as possible in order to

catch the main picture. The next step is to assign each data point to the nearest cluster by comparing the distance to respective centroids and calculate new centroids based on the resulting clusters. After we have these new centroids, the procedure should be repeated until no more changes are done regarding the new centroids. Besides K-means, there exists a model based approach which uses a certain model as clusters and attempts to fit the data using this model. In practice, some parametric distributions are employed as the model for each cluster, e.g. Gaussian distribution for continuous cases or Poisson distribution for discrete cases. Therefore a mixture model is used to model the entire data set. An appropriate mixture model should have the properties that first, data in each cluster should be tight and distributions for each component should have high peaks; secondly the dominant pattern of the data should be well captured by this mixture model. A Gaussian Mixture Model (GMM) are commonly used to fit continuous distributions.

However, these methods seem to only use samples and disregard the fact that the density is sometimes also available. In this chapter, we propose a new method to fit the distribution by mixture models via minimizing L_p norm which involves the usage of both samples and densities. It is obvious that the algorithm for minimizing L_p has consistency property unlike maximum likelihood estimate. Moreover, the estimate is stable and unaffected by outliers especially when p is large. Furthermore, the mixture model $f_\theta(x)$ could be any parametric form besides mixture of Gaussian. The advantage of employing mixture of Gaussian in the experiments here is that we could use the means of Gaussian as our clusters. Therefore, the Minimum L_p Norm Estimator is very general and different forms of $f_\theta(x)$ and different values of p would help discover different characterizations of the distribution.

4.1 Clustering via L_p distance - the Minimum L_p Norm Estimator

As mentioned in introduction, many sampling procedures employed for inference in high dimensional spaces return the posterior distribution $f(x)$ (or subject to a normalization constant) of each observation as well as the sampled values, e.g. MCMC sampling or importance sampling. However, it is difficult to visualize or characterize high-D spaces.

Clustering is commonly a first step to reveal the properties of such high-D distributions. There are hundreds of clustering methods. I focus on model based clustering in this chapter, in which mixture models are fit to samples, with each mixing component represents a cluster, e.g. Gaussian Mixture Model (GMM). Given samples, it is common to employ maximum likelihood estimation to learn parameters from training data by implementing the iterative Expectation-Maximization (EM) algorithm. Here, I consider the use of density function of each observation, to ask if it is possible to improve on MLE.

To answer these questions, we propose a new approach to estimate parameters based on minimizing L_p norm as follows:

Minimum L_p Norm Estimator:

$$\Theta = \text{argmin}_\theta E|f(x) - f_\theta(x)|^p \tag{4.1}$$

where θ is the parameters for the component distributions, and the expectation is under the probability distribution with density function $f(x)$.

Remark:

1. In particular, if $p = 2$

$$\begin{aligned}\Theta &= \text{argmin}_\theta E|f(x) - f_\theta(x)|^2 \\ &= \text{argmin}_\theta \left(-2E[f(x)f_\theta(x)] + Ef_\theta(x)^2\right) \\ &= \text{argmax}_\theta \left(\langle f(x), f_\theta(x)\rangle_f - 0.5||f_\theta(x)||^2_{L_2(f)}\right).\end{aligned}$$

   Thus, minimizing the L_2 distance is equivalent to maximizing the inner product $\langle f(x), f_\theta(x)\rangle_f$ minus half of the L_2 norm $0.5||f_\theta(x)||^2_{L_2(f)}$ which can be viewed as a regularization term.

2. As the number of mixing components of f_θ, m, tends to infinite,

$$\min_\theta E|f(x) - f_\theta(x)|^p \longrightarrow 0.$$

3. There are many minimizers in general, since the labels of mixing components can be arbitrarily permuted while maintain the density f_θ. Thus, we can think of Θ as the set of all minimizers instead of a single optimal solution.

4. This estimator has consistency property in the sense that if f is truly a mixture of corresponding component distributions, then the Θ includes the true parameter of f.

5. The mixing component "f_θ" does not need to be Gaussian. It could be any appropriate parametric form in principal. For discrete cases, we replace the densities f and f_θ with mass functions p and p_θ.

4.1.1 Theoretical Study on choices of "p"

Although we proposed the Minimum L_p Norm Estimator in (4.1), we still face the question of the choice of p. Will different values of p give different estimates and what is the difference and relation between these different estimators? After exploring these questions,we may choose different p depending on applications. First, let us

investigate the limiting phenomena of p. To start, we introduce the definition for a new convergence behavior as follows.

Definition: Let A_n's and A be compact sets and d be corresponding metric in the space. We say A_n tends into A,

$$A_n \hookrightarrow A,$$

if as n tends infinity,

$$\max_{a \in A_n}(\min_{b \in A} d(a,b)) \to 0.$$

With this definition, we are able to explore the limiting behavior of the Minimum L_p Norm Estimator. We have the following theorem.

Theorem 4.1.1. *Let the parameter space be compact and f_θ be smooth with respect to θ under sup-norm, then*

(1) as p tends to zero,

$$\text{argmin}_\theta E|f(x) - f_\theta(x)|^p \hookrightarrow \text{argmax}_\theta Pro(\{x : f(x) = f_\theta(x)\})$$

(2) as p tends to infinity,

$$\text{argmin}_\theta E|f(x) - f_\theta(x)|^p \hookrightarrow \text{argmin}_\theta \left(\sup_x |f(x) - f_\theta(x)| \right)$$

Proof. (1) Let $\Theta_0 = \text{argmax}_\theta Pro(\{x : f(x) = f_\theta(x)\})$. Assume $\text{argmin}_\theta E|f(x) - f_\theta(x)|^p$ does not tend into $\text{argmax}_\theta Pro(\{x : f(x) = f_\theta(x)\})$. Then by definition, there exists an δ and a sequence θ_{p_k} such that $\min_{\theta \in \Theta_0} d(\theta_{p_k}, \theta) > \delta$ for all p_k. Since we assume the parameter space are compact, therefore, there is a convergence subsequence of $\{\theta_{p_k}\}_{k=1}^\infty$. For simplification, let us still call them $\{\theta_{p_k}\}_{k=1}^\infty$, and let the limit $\lim_{k \to \infty} \theta_{p_k} = \phi$. Then we have $\min_{\theta \in \Theta_0} d(\phi, \theta) \geq \delta$. Thus ϕ is not in Θ_0 and $Pro(\{x : f(x) = f_\phi(x)\}) < \max_\theta Pro(\{x : f(x) = f_\theta(x)\})$. Now, let $\max_\theta Pro(\{x : f(x) = f_\theta(x)\}) - Pro(\{x : f(x) = f_\phi(x)\}) = \epsilon$.

Then selecting a θ_0 from the set Θ_0, we know

$$E|f(x) - f_{\theta_0}(x)|^p \to 1 - Pro(\{x : f(x) = f_{\theta_0}(x)\}),$$

as $p \to 0$, and

$$E|f(x) - f_{\theta_{p_k}}(x)|^{p_k} \to 1 - Pro(\{x : f(x) = f_{\phi}(x)\}),$$

as k goes to infinity. Therefore, when k large enough,

$$E|f(x) - f_{\theta_0}(x)|^{p_k} < 1 - Pro(\{x : f(x) = f_{\theta_0}(x)\}) + \frac{1}{3}\epsilon.$$

and

$$E|f(x) - f_{\theta_{p_k}}(x)|^{p_k} > 1 - Pro(\{x : f(x) = f_{\phi}(x)\}) - \frac{1}{3}\epsilon.$$

Hence,

$$\begin{aligned} E|f(x) - f_{\theta_0}(x)|^{p_k} &< 1 - Pro(\{x : f(x) = f_{\theta_0}(x)\}) + \frac{1}{3}\epsilon \\ &= 1 - Pro(\{x : f(x) = f_{\phi}(x)\}) - \frac{2}{3}\epsilon \\ &< E|f(x) - f_{\theta_{p_k}}(x)|^{p_k}, \end{aligned}$$

and we get contradiction with definition of θ_{p_k}. The proof of the first part is complete.

(2) The proof of second part is essentially the same as the first part by using the fact that

$$(E|f(x) - f_\theta(x)|^p)^{\frac{1}{p}} \to \sup_x |f(x) - f_\theta(x)|$$

as $p \to \infty$.

Let $\Theta_0 = \text{argmin}_\theta \left(\sup_x |f(x) - f_\theta(x)|\right)$. Assume $\text{argmin}_\theta E|f(x) - f_\theta(x)|^p$ does not tend into $\text{argmin}_\theta \left(\sup_x |f(x) - f_\theta(x)|\right)$. Then by definition, there exists an δ and a sequence θ_{p_k} such that $\min_{\theta \in \Theta_0} d(\theta_{p_k}, \theta) > \delta$ for all p_k. Since we

assume the parameter space are compact, therefore, there is a convergence subsequence of $\{\theta_{p_k}\}_{k=1}^{\infty}$. For simplification, let us still call them $\{\theta_{p_k}\}_{k=1}^{\infty}$, and let the limit $\lim_{k\to\infty}\theta_{p_k} = \phi$. Then we have $\min_{\theta\in\Theta_0} d(\phi,\theta) \geq \delta$. Thus ϕ is not in Θ_0 and $(\sup_x |f(x) - f_\phi(x)|) > \min_\theta (\sup_x |f(x) - f_\theta(x)|)$. Now, let $(\sup_x |f(x) - f_\phi(x)|) - \min_\theta (\sup_x |f(x) - f_\theta(x)|) = \epsilon$. Then selecting a θ_0 from the set Θ_0, we know

$$(E|f(x) - f_{\theta_0}(x)|^p)^{\frac{1}{p}} \to \sup_x |f(x) - f_{\theta_0}(x)|$$

as $p \to \infty$, and

$$(E|f(x) - f_{\theta_{p_k}}(x)|^p)^{\frac{1}{p}} \to \sup_x |f(x) - f_\phi(x)|$$

as k goes to infinity. Therefore, when k large enough,

$$E(|f(x) - f_{\theta_0}(x)|^{p_k})^{\frac{1}{p_k}} < \sup_x |f(x) - f_{\theta_0}(x)| + \frac{1}{3}\epsilon.$$

and

$$E(|f(x) - f_{\theta_{p_k}}(x)|^{p_k})^{\frac{1}{p_k}} > \sup_x |f(x) - f_\phi(x)| - \frac{1}{3}\epsilon.$$

Hence,

$$\begin{aligned} E(|f(x) - f_{\theta_0}(x)|^{p_k})^{\frac{1}{p_k}} &< \sup_x |f(x) - f_{\theta_0}(x)| + \frac{1}{3}\epsilon \\ &= \sup_x |f(x) - f_\phi(x)| - \frac{2}{3}\epsilon \\ &< E(|f(x) - f_{\theta_{p_k}}(x)|^{p_k})^{\frac{1}{p_k}}, \end{aligned}$$

and we get contradiction with definition of θ_{p_k}.

□

Remark: Theorem 4.1.1 demonstrates the following fitting phenomena:

(1) When p is small, the Minimum L_p Norm Estimator tries to fit all samples with

little regard for the probabilities or density values of the sampled values.

(2) When p is large, the Minimum L_p Norm Estimator tries to fit the samples with higher probabilities or higher densities.

This phenomenon gives us flexibilities when making applications. For some problems, we may want to focus on high probability samples, but for others, we may want to consider it more globally. With Theorem 4.1.1, we could choose different values of p to meet our purpose.

4.1.2 Empirical L_p distance

In the optimization to obtain the Minimum L_p Norm Estimator (4.1), the L_p norm

$$E|f(x) - f_\theta(x)|^p = \int |f(x) - f_\theta(x)|^p f(x)dx$$

may be difficult to compute, especially in high dimensional space. Although in some situations dynamic programming might help in discrete cases. Alternatively we could approximate the expectation by law of large number, or through the use of Monte Carlo method as follows:

Empirical L_p distance:

$$E_{em}|f(x) - f_\theta(x)|^p = \frac{1}{N}\sum_{i=1}^{N} |f(x_i) - f_\theta(x_i)|^p$$

where x_i's are i.i.d samples from f. Therefore, we have an alternative estimator

$$\Theta = \text{argmin}_\theta \frac{1}{N}\sum_{i=1}^{N} |f(x_i) - f_\theta(x_i)|^p \tag{4.2}$$

which involves both samples and densities of the samples.

In a complicated system with high dimension random vectors, if we have samples and their distributions (density function values or mass function values), this

empirical L_p norm provides a computable approximation to estimate the parameter θ using both the sampled value and the probabilities or probability densities. Given samples x_i's to estimate the parameters, the most common method is to use maximum likelihood estimator(MLE). However, the MLE is only based on the samples x_i's and the probabilities of the samples $p(x_i)$'s remain unused. In next section, we will demonstrate the utilities of this empirical L_p estimator by comparison with MLE.

Besides L_p Norm distance, there are many other measurements, which can be approximated empirically involving both samples and their distributions. For example consider the case of variational Bayesian methods.

Variational Bayesian methods Variational Bayesian methods, also called "ensemble learning", are a family of techniques for approximating integrals. They are commonly used in complicated statistical models where there exist observed variables as well as latent variables and unknown parameters. Variational Bayesian method is essentially to minimize the relative entropy $D(Q||P)$ where P is the true distribution and Q is a simpler distribution like mixture Gaussian. Now, we use importance sampling to get the approximation:

$$D(Q||P) = \int Q(x) log \frac{Q(x)}{P(x)} dx \sim \frac{1}{N} \sum_{i=1}^{N} \frac{Q(x_i)}{P(x_i)} log \frac{Q(x_i)}{P(x_i)},$$

where x_i's are i.i.d sample from the true distribution P.

Consider minimizing $D(P||Q)$ instead of $D(Q||P)$. Then, minimizing the sampling approximation is equivalent to MLE:

$$\min D(P||Q) = \min \left[\int P(x) log \frac{P(x)}{Q(x)} dx \right] \sim \min \left[\sum_{i=1}^{N} log \frac{P(x_i)}{Q(x_i)} \right] \sim \max \left[\prod_{i=1}^{N} Q(x_i) \right]$$

4.2 Algorithms

Specifically, consider Gaussian mixture model (GMM),

$$f(x; \{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m) = \sum_{i=1}^m \epsilon_i N(x; \mu_i, \Sigma_i). \tag{4.3}$$

Given the parameters $\{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m$, the i-th cluster can be labelled by its mean μ_i, and the covariance matrix Σ_i reports the degree of concentration and the correlations among variables in the i-th cluster. ϵ_i represents the proportion of population of the i-th cluster.

In this section we give an algorithm which fits the distribution with mixture of Gaussian by minimizing the empirical L_p norm as proposed in the Section 4.1.

Minimum L_p Norm Estimator:

$$\Theta = \text{argmin}_\theta \frac{1}{N} \sum_{i=1}^N |f(x_i) - f_\theta(x_i)|^p \tag{4.4}$$

where $\theta = \{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m$, $f_\theta(x) = f(x; \{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m)$ as in (4.3).

4.2.1 Modified Gradient Decent Algorithms

Let us denote the goal minimization equation as

$$T = \frac{1}{N} \sum_{i=1}^N |f(x_i) - f_\theta(x_i)|^p. \tag{4.5}$$

Before we give the algorithm, let us review some matrix calculus which would be helpful in the derivation of the gradient for these parameters.

Notation: Let $M(n, m)$ denote the space of real $n \times m$ matrices with n rows and m columns, such matrices will be denoted using bold capital letters: $\mathbf{A}$, $\mathbf{X}$, etc. An element of $M(n, 1)$, that is, a column vector, is denoted with a boldface lowercase letter: $\mathbf{a}$, $\mathbf{x}$, etc. An element of $M(1, 1)$ is a scalar, denoted with lowercase

typeface: a, t, x, etc. $\mathbf{X}^T$ denotes matrix transpose, $\mathrm{tr}(\mathbf{X})$ is trace, and $\det(\mathbf{X})$ is the determinant. All functions are assumed to be of differentiability class.

Vector calculus:

- The gradient of a scalar function $f : \mathbb{R}^n \to \mathbb{R}$
  $\frac{\partial f}{\partial \mathbf{x}} = [\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}]$

Matrix calculus:

- The gradient of a scalar function $f : M(n,m) \to \mathbb{R}$
  $$\frac{\partial f}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial f}{\partial X_{1,1}} & \cdots & \frac{\partial f}{\partial X_{n,1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial X_{1,m}} & \cdots & \frac{\partial f}{\partial X_{n,m}} \end{bmatrix}$$

Examples:

- $\frac{\partial \mathbf{A}^T\mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T\mathbf{A}}{\partial \mathbf{x}} = \mathbf{A}$
- $\frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}$
- $\frac{\partial (\mathbf{A}\mathbf{x}+\mathbf{b})^T\mathbf{C}(\mathbf{D}\mathbf{x}+\mathbf{e})}{\partial \mathbf{x}} = (\mathbf{D}\mathbf{x}+\mathbf{e})^T\mathbf{C}^T\mathbf{A} + (\mathbf{A}\mathbf{x}+\mathbf{b})^T\mathbf{C}\mathbf{D}$
- $\frac{\partial tr(\mathbf{A}\mathbf{X})}{\partial \mathbf{X}} = \frac{\partial tr(\mathbf{X}\mathbf{A})}{\partial \mathbf{X}} = \mathbf{A}$
- $\frac{\partial \det \mathbf{X}}{\partial \mathbf{X}} = \det \mathbf{X} \cdot \mathbf{X}^{-1}$

Now we are ready to give the following algorithm.

Algorithm 4.2.1: *The Minimum L_p Norm Estimator.*

In order to minimize (4.5), iterate the following two steps until convergence:

1. *Use gradient descent to minimize (4.5) over ϵ_j's and μ_j's.*
2. *Use gradient descent to minimize (4.5) over Σ_j's under the constraint that Σ_j's are positive definite.*

Remark: In step two the constraint that Σ_j's are positive definite is important since Σ_j's are the covariance matrix in Gaussian components and the covariance matrix must be positive definite.

Formulas:

In step one, we first calculate the corresponding gradient for ϵ_j's and μ_j's. The derivative with respect to ϵ_j is easy to obtain as

$$\frac{\partial T}{\partial \epsilon_j} = -\sum_{i=1}^{N} p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} N(x_j; \mu_j, \Sigma_j). \quad (4.6)$$

As to the derivative with respect to μ_j, we first have

$$\frac{\partial T}{\partial \mu_j} = -\sum_{i=1}^{N} p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} \epsilon_j \frac{\partial N(x_i; \mu_j; \Sigma_j)}{\partial \mu_j}. \quad (4.7)$$

We could write out the density functions for Gaussian distribution as

$$\begin{aligned} N(x_i; \mu_j; \Sigma_j) &= \frac{1}{(2\pi)^{\frac{d}{2}}} \det(\Sigma_j)^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\Big) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}}} \det(\Sigma_j)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} tr\Big((x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\Big)\right) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}}} \det(\Sigma_j)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} tr\Big((x_i - \mu_j)(x_i - \mu_j)^T \Sigma_j^{-1}\Big)\right). \end{aligned}$$

Then we have

$$\ln N(x_i; \mu_j; \Sigma_j) = const - \frac{1}{2}\ln\det(\Sigma_j) - \frac{1}{2} tr\Big((x_i - \mu_j)(x_i - \mu_j)^T \Sigma_j^{-1}\Big).$$

The differentiation of this log-likelihood is

$$\begin{aligned} d\ln N(x_i; \mu_j; \Sigma_j) = &-\frac{1}{2} tr\Big[\Sigma_j^{-1}\{d\Sigma_j\}\Big] - \frac{1}{2} tr\Big[-\Sigma_j^{-1}\{d\Sigma_j\}\Sigma_j^{-1}(x_i - \mu_j)(x_i - \mu_j)^T \\ &- 2\Sigma_j^{-1}(x_i - \mu_j)\{d\mu_j\}^T\Big]. \end{aligned} \quad (4.8)$$

Therefore,

$$\frac{\partial \ln N(x_i;\mu_j;\Sigma_j)}{\partial \mu_j} = \Sigma_j^{-1}(x_i-\mu_j),$$

$$\frac{\partial N(x_i;\mu_j;\Sigma_j)}{\partial \mu_j} = N(x_i;\mu_j;\Sigma_j)\Sigma_j^{-1}(x_i-\mu_j).$$

Then from Equation (4.7), we could obtain

$$\frac{\partial T}{\partial \mu_j} = -\sum_{i=1}^{N} p * \Big(sign\Big(f(x_i)-f_\theta(x_i)\Big)\Big)^p \Big(f(x_i)-f_\theta(x_i)\Big)^{p-1} * \epsilon_j * N(x_i;\mu_j;\Sigma_j)\Sigma_j^{-1}(x_i-\mu_j). \tag{4.9}$$

With (4.6) and (4.9), we could employ gradient descent method to minimize the goal function (4.5) over ϵ_i's and μ_i's.

Now we turn to step 2 in Algorithm **4.2.1**. The derivative of the goal function (4.5) with respect to Σ_j is

$$\frac{\partial T}{\partial \Sigma_j} = -\sum_{i=1}^{N} p * \Big(sign\Big(f(x_i)-f_\theta(x_i)\Big)\Big)^p \Big(f(x_i)-f_\theta(x_i)\Big)^{p-1} \epsilon_j \frac{\partial N(x_i;\mu_j;\Sigma_j)}{\partial \Sigma_j}. \tag{4.10}$$

From the differentiation form (4.8), the differentiation of this log-likelihood with respect to Σ_j is

$$\begin{aligned} d\ln N(x_i;\mu_j;\Sigma_j) &= -\frac{1}{2}tr\Big[\Sigma_j^{-1}\{d\Sigma_j\}\Big] - \frac{1}{2}tr\Big[-\Sigma_j^{-1}\{d\Sigma_j\}\Sigma_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T\Big] \\ &= -\frac{1}{2}tr\Big[\Sigma_j^{-1}\{d\Sigma_j\}\Big(I-\Sigma_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T\Big)\Big] \\ &= -\frac{1}{2}tr\Big[\Big(I-\Sigma_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T\Big)\Sigma_j^{-1}\{d\Sigma_j\}\Big]. \end{aligned} \tag{4.11}$$

Since

$$\frac{\partial \ln N(x_i;\mu_j;\Sigma_j)}{\partial \Sigma_j} = -\frac{1}{2}\Big(I-\Sigma_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T\Big)\Sigma_j^{-1}, \tag{4.12}$$

Then,

$$\frac{\partial N(x_i;\mu_j;\Sigma_j)}{\partial \Sigma_j} = -\frac{1}{2}N(x_i;\mu_j;\Sigma_j)\Big(I-\Sigma_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T\Big)\Sigma_j^{-1}. \tag{4.13}$$

At last, from equation (4.10), we have

$$\frac{\partial T}{\partial \Sigma_j} = \sum_{i=1}^{N} p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} \epsilon_j \frac{1}{2} N(x_i; \mu_j; \Sigma_j) \Big(I - \Sigma_j^{-1}(x_i - \mu_j)(x_i - \mu_j)'\Big)\Sigma_j^{-1}. \tag{4.14}$$

Remark: Actually by the differentiation form of the log-likelihood, we are able to derive the explicit form of minimizers for Σ_j's, which is

$$\frac{\sum_{i=1}^{N}(x_i - \mu_j)(x_i - \mu_j)^T \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} N(x_i; \mu_j; \Sigma_j)}{\sum_{i=1}^{N} \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} N(x_i; \mu_j; \Sigma_j)}.$$

However since the coefficient terms $\Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1}$ are not always positive, we could not guarantee that these Σ_j's are positive definite, which is the prerequisite for being covariance matrix of a Gaussian distribution. This is the reason we choose to employ the gradient descent technique in Algorithm **4.2.1**. In this algorithm, in order to look for the local minimum of the goal function (4.5), we start with a positive definite matrix Σ_j and similar to gradient descent we take steps proportional to the negative of the gradient of the function at the current point. However, we carefully take the step small enough such that the resulting new matrix is still positive definite. In summary, the pseudo codes of our modified gradient descent method for deriving new Σ_j's is as follows.

Pseudo Codes of modified gradient descent method for Σ_j's:

Until convergence, do

{

h =some threshold;

do

{

$\Sigma_j^{new} = \Sigma_j - h * \frac{\partial T}{\partial \Sigma_j}$ *for* $j = 1, \cdots, m$;

$h = \frac{h}{2}$;

}

Until all the Σ_j's are positive definite.

}

4.2.2 The case of lacking partition function

In practice, we often encounter the situations when the partition function or normalization constant Z of distributions is unknown. For MLE, it does not matter since MLE only uses the samples drawn from the underlying distribution and does not depend on the distributions or densities of these samples. However the circumstance is different for our empirical L_p estimator (4.2). Consider L_p distance of non-normalized distribution $p(x)$ and $zp_\theta(x)$, where z is another unknown parameter regarded as its normalization constant of $p(x)$. Then, in these situations our estimator will be obtained by minimizing the empirical L_p norm

$$E_{em}|p(x) - zp_\theta(x)|^p = \frac{1}{N}\sum_{i=1}^{N}|p(x_i) - zp_\theta(x_i)|^p. \tag{4.15}$$

In this estimate (4.15), in addition to the parameters in the components of the mixing component distributions, there is one more parameter corresponding to the missing normalization constant, i.e. z. Now we could iteratively estimate the parameter θ and z. For discrete cases, we could initialize z by $\sum_{x\in A} p(x)$ which is smaller than the real Z and tends to over fit the samples, where A is the support of samples. For continuous cases, we could, for example, initialize z by approximating the integral $\int p(x)dx$(depending on applications). We will present more details in experiment 5 and experiment 6 in Section 4.3. In particular, for $p = 2$ and given θ, the maximizer is

$$z_{max} = \frac{\sum_{i=1}^{N} p(x_i)}{\sum_{i=1}^{N} p_\theta(x_i)}.$$

Moreover, this new estimator has the same properties as in Remark 2 and 4 in Section 4.1. This new estimator also has similar limiting behavior as in Theorem 4.1.1.

In such cases where normalization constant is hard to achieve, our algorithm

could be modified a bit based on Algorithm **4.2.1** and is then aiming to minimize

$$\Theta = \text{argmin}_{\theta,z} \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - z f_\theta(x_i)|^p, \tag{4.16}$$

where $\theta = \{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m$, $f_\theta(x) = f(x; \{\epsilon_i, \mu_i, \Sigma_i\}_{i=1}^m)$.

In (4.16), we have one more parameter z for the absent normalization constant. Therefore in each iteration of our algorithm, in addition to update μ_j's, ϵ_j's and Σ_j's as in Algorithm **4.2.1**, we need to update z.

Algorithm 4.2.2: *The Minimum L_p Norm Estimator in absence of Normalization Constant.*

1. *Use gradient descent to minimize (4.16) over ϵ_j's and μ_j's.*
2. *Use gradient descent to minimize (4.16) over Σ_j's under the constraint that Σ_j's are positive definite*
3. *Use gradient descent to minimize (4.16) over z.*

The formulas for the gradient could be obtained by similar procedures as before. **Formulas for the partial derivatives:**

$$\begin{aligned}
\frac{\partial T}{\partial \epsilon_j} =& -\sum_{i=1}^{N} z * p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} N(x_j; \mu_j, \Sigma_j) \\
\frac{\partial T}{\partial \mu_j} =& -\sum_{i=1}^{N} z * p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} \\
& * \epsilon_j * N(x_i; \mu_j; \Sigma_j) \Sigma_j^{-1} (x_i - \mu_j) \\
\frac{\partial T}{\partial \Sigma_j} =& \sum_{i=1}^{N} z * p * \Big(sign\Big(f(x_i) - f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} \epsilon_j \frac{1}{2} N(x_i; \mu_j; \Sigma_j) \\
& \Big(I - \Sigma_j^{-1}(x_i - \mu_j)(x_i - \mu_j)'\Big) \Sigma_j^{-1} \\
\frac{\partial T}{\partial z} =& -\sum_{i=1}^{N} p * \Big(sign\Big(f(x_i) - z f_\theta(x_i)\Big)\Big)^p \Big(f(x_i) - f_\theta(x_i)\Big)^{p-1} f_\theta(x_i)
\end{aligned}$$

Remark: To avoid local maxima, we'd better choose appropriate initial values to start our iteration. For μ_j's, ϵ_j's and Σ_j's, we have several different approaches to get an estimate to start with, e.g. using the result from standard EM algorithm as the starting point. Here we provide an approach to obtain an estimate for the normalization constant z.

For d dimensional space, we partition each dimension into k even subintervals and in total there are k^d grids $I_1,\ I_2, \cdots, I_{k^d}$. Let $A = \text{unique } (\mathbf{x}_1^N)$, for nonempty grid I_j define

$$E(j) = \frac{1}{n(j)} \sum_{\mathbf{x} \in \{\mathbf{x}_1^N\} \bigcap I_j} f(\mathbf{x}) |I_j|, \tag{4.17}$$

where $n(j)$ is the number of data points in grid I_j, $|I_j|$ is the volume of this grid, and $f(j)$ is corresponding density subject to the normalization constant.

Then our estimate for the normalization constant is $z = \sum_{j=1}^{k^d} E(j)$.

In summary, in this section we provide algorithms and formulas for the derivation of the Empirical Minimum L_p Estimator. In the following section we will present the application of these algorithms to some examples and illustrate the findings.

4.3 Experiments illustrating behavior of the Minimum L_p Norm Estimator

In this section, we illustrate by examining a mixture Gaussian distribution with ten components, which has three big modes and several small modes (Figure 4.1). We perform Algorithm **4.2.1** and Algorithm **4.2.2** to fit this distribution with various L_p norm and various number of fitting modes, and compare the results. We also discuss the convergence behavior of our algorithm and compare it with K-means and Maximum Likelihood Estimator.

Experiment 1: Fitting using different number of modes In this experiment, we perform Algorithm **4.2.1** to minimize the L_2 norm with various number of fitting modes. Figure 4.2 gives the fitting result using number of modes from one to six

and Figure 4.3 gives corresponding L_2 norm measure. In Figure 4.3, it is obvious that there is a huge decrease of L_2 norm when we fit by three mode Gaussian and afterward the L_2 norm seems to decrease gently until convergence. Indeed, there are three big modes in the true distribution and our algorithm does locate them: the means of the three mode Gaussian by our algorithm correspond to the means of the three big modes in the true distribution. Furthermore, in Figure 4.2 we could see that the resulting mixture model is getting closer to the true distribution as the number of fitting modes increases. If we use ten mode model to fit it, we are able to recover the exact distribution, which verifies our assertion in Section **4.2** that the Minimum L_p Norm Estimator is consistent.

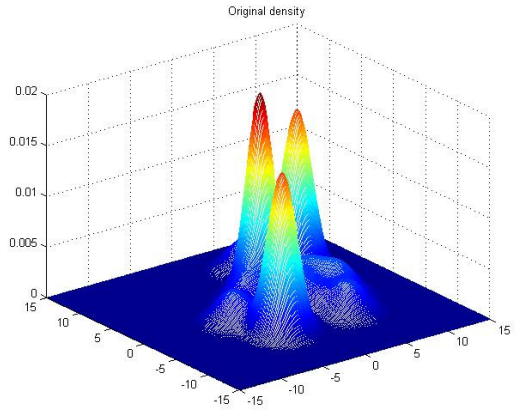
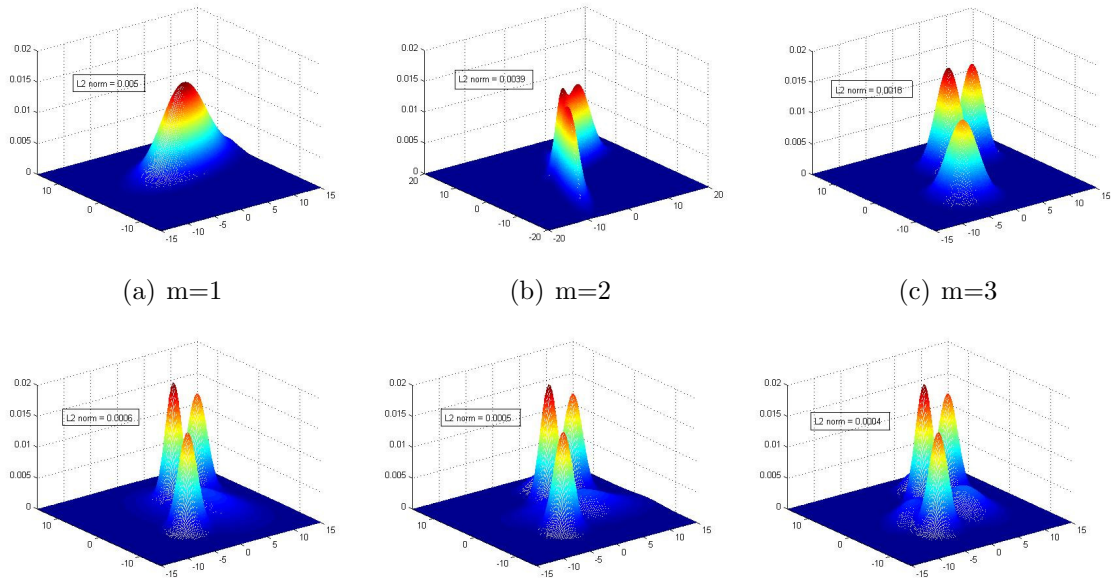


Figure 4.1: Original density for a ten mode Gaussian Mixture

Experiment 2: Fitting using different values of p In this experiment, we examine the behavior with various L_p norm minimization. Figure 4.4 gives the fitting results for p=1, 2, 3, 4, 5, 10. In Figure 4.4 we could see that when p is small, the resulting mixture model tends to cover more ensemble in the space. As p gets larger, the resulting mixture model has more concentrate peaks around the mean. This is again a verification of Theorem **4.1.1**.



(a) m=1 (b) m=2 (c) m=3

(d) m=4 (e) m=5 (f) m=6

Figure 4.2: **Mixture of Gaussian fitting.** Using one to six mixtures of Gaussian to fit the original density. Corresponding L_2 norm are 0.005, 0.0039, 0.0018, 0.0006, 0.0005, 0.0004

Experiment 3: Speed of Convergence In this experiment we examine the convergence behavior of our algorithm. We showed that our algorithm is consistent and here we use three component Gaussian to fit a true three mode Gaussian distribution. It is obvious that the minimizer in this situation would definitely replicate the true distribution since with the true parameters the corresponding L_p norm is zero. Figure 4.5 gives the fitting model after five and twenty iterations. This algorithm converges fast in that as to the fifth iteration, the mixture model we found is already very close to the true distribution. It could also be verified by checking the trend of L_2 norm in Figure 4.6 which shows very fast decrease of L_2 norm.

Experiment 4: Comparison with K-means and MLE In this experiment we compare the result of our algorithm with those of K-means and Maximum Likelihood Estimator (MLE). We still illustrate by the same ten components Gaussian distribution and try to do three mode fitting using these three different methods. Figure 4.7 displays the resulting models from these three methods, which are very

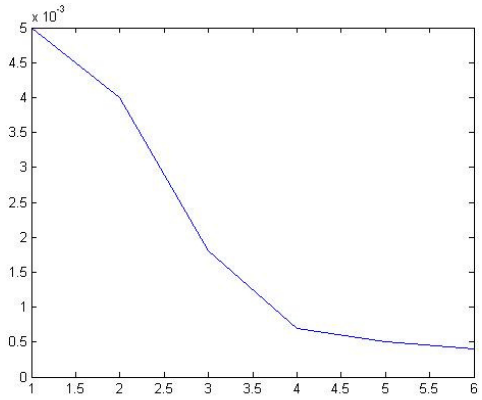
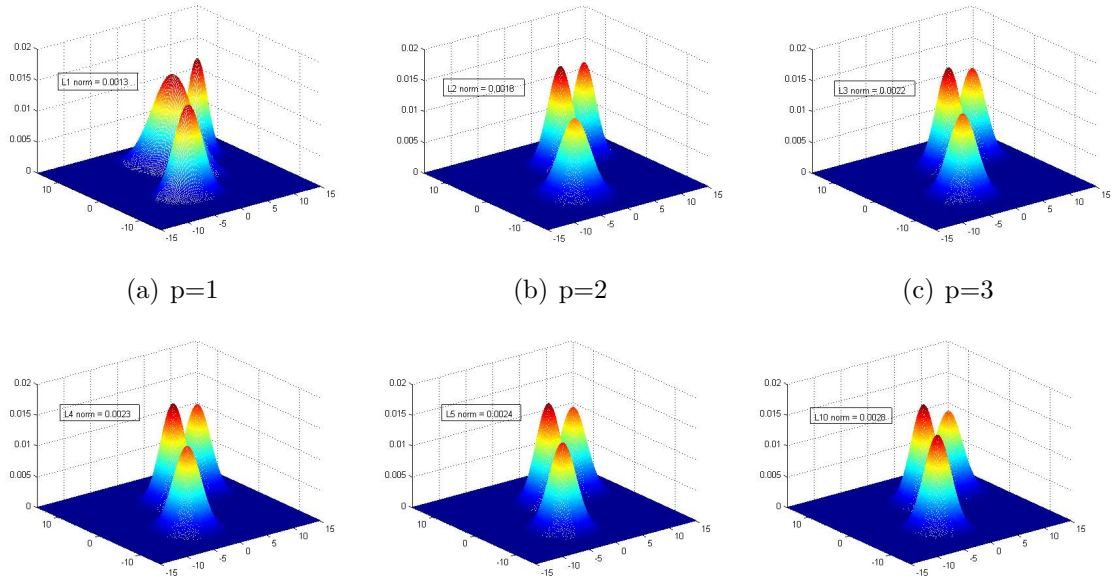


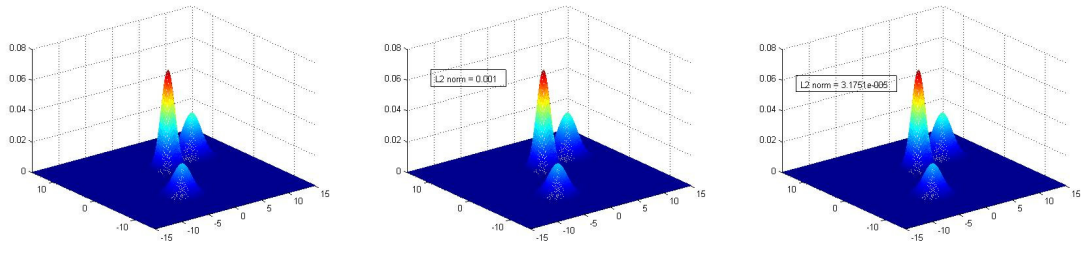
Figure 4.3: **Trend of corresponding L_2 norm.** L_2 norm for fitting models with number of component one to six



(a) p=1 (b) p=2 (c) p=3

(d) p=4 (e) p=5 (f) p=10

Figure 4.4: **Three mode mixture of Gaussian fitting with various L_p norm.** For p=1, 2, 3, 4, 5, 10, corresponding L_2 norm are 0.0013, 0.0018, 0.0022, 0.0023, 0.0024, 0.0028



(a) original (b) after 5 iterations (c) after 20 iterations

Figure 4.5: **Convergence behavior.** First figure is a three mode Gaussian density; The Second and third is the parameter fitting after 5th and 20th iteration; The last one is the L_2 norm behavior as the iteration runs

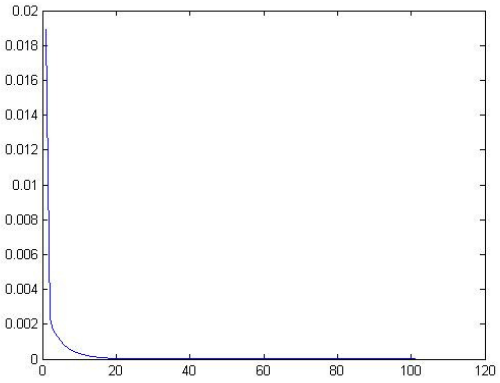
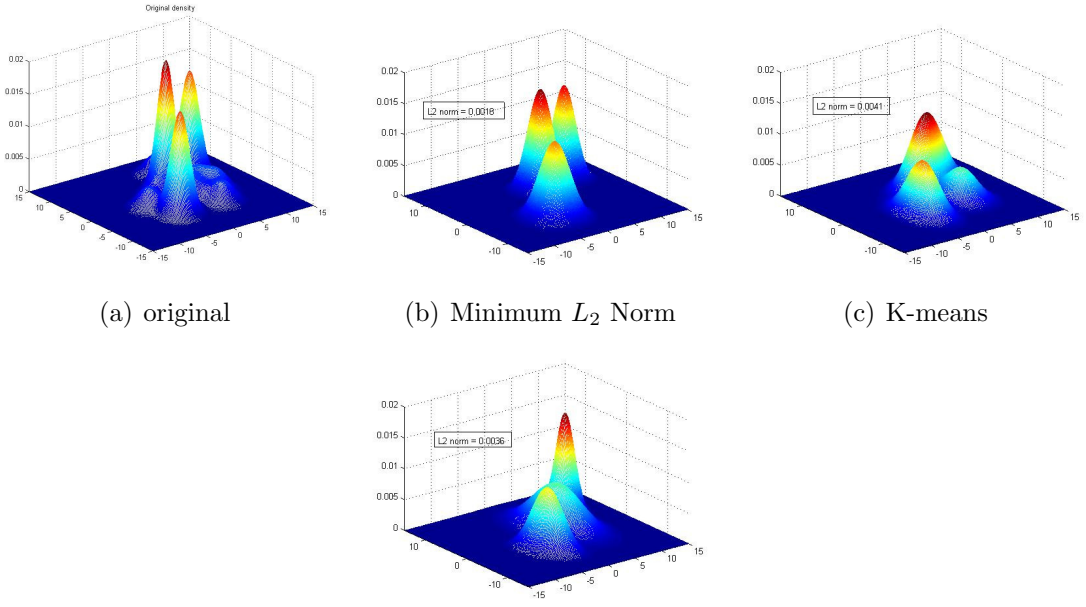


Figure 4.6: L_2 norm after each iteration

different from each other. However, our algorithm is the unique one which is able to locate the means of the three big modes in the true distribution. For the other two methods, they somehow tend to put more attention on the tails and hence have the peaks in the middle. In this sense, our algorithm is able to catch the main modes of some density with noise. In contrast, K-means and MLE seem to catch some other features of the true distribution, however they pay more attention to the main mass distribution.



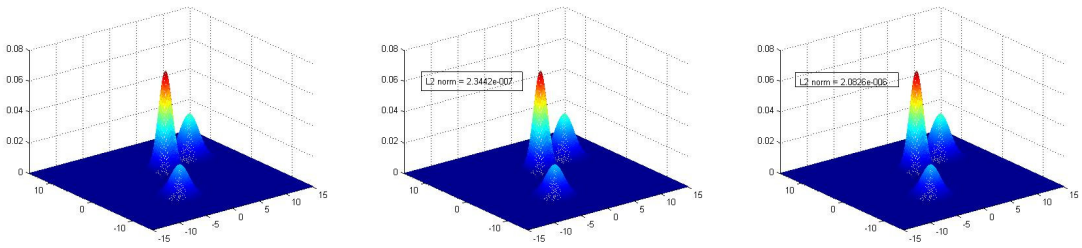
(a) original (b) Minimum L_2 Norm (c) K-means

(d) MLE

Figure 4.7: **Comparison with K-means and MLE.** Figure (a) is the original ten mode Gaussian density; Figure (b) is the three mode L_2 norm fitting using our algorithm; The parameters of Figure (c) is estimated by clustering using K-means; The parameters of Figure (d) is the MLE solution to three mode Gaussian fitting

Experiment 5: In absence of normalization constant In the following two experiments, our experiments are focusing on samples with their proportion of probabilities instead of exact probabilities (subject to a normalization constant). First we work on three components mixture of Gaussian distribution and we multiply their density by some constant and see whether our algorithm is able to find the

true parameters and this multiplier. We use a three model mixture model to fit this distribution and perform Algorithm **4.2.2**. Figure 4.8 displays the result when the multiplier is 5 and 12 respectively. Even we only have the densities for the samples subject to a normalization constant, Algorithm **4.2.2** is still able to recover the original density well and find the normalization constant. This might be an approach from which people could get an estimate of the normalization constant from samples and their densities subject to this normalization constant.

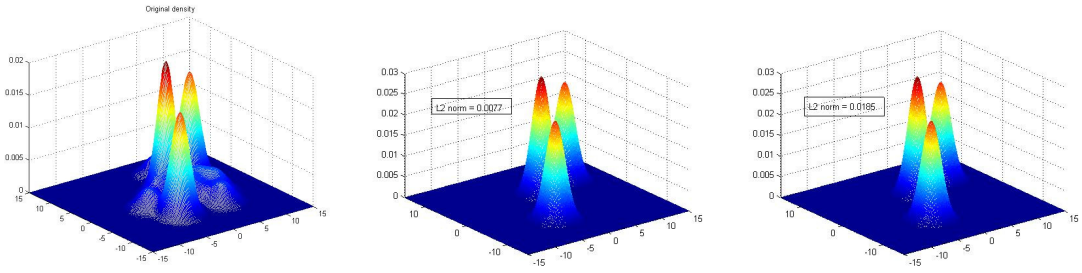


(a) original (b) multiply by 5 (c) multiply by 12

Figure 4.8: **Mixture of Gaussian fitting with density subject to a constant.** Figure (a) is the density which is composed of three Gaussian; Figure (b) and (c) are the fitting result for normalization constant 5 and 12.

Experiment 6: Effect of initial values for normalization constant In this experiment we are working on the same clustering experiment as in Experiment 1, while multiplying a constant to the true densities, i.e. we assume that we don't have the exact densities for the samples but only those subject to a normalization constant. During this experiment, we found that an appropriate starting value for the normalization constant term in EM algorithm is very important. We tried a set of different initial values and as shown in Figure 4.10 and Figure 4.11, with small to moderate initial values, the algorithm would easily converge to the true mixture of Gaussian distribution. However, if we start with a large staring value, it usually doesn't converge to the ideal one but to another local maxima. While it might worth paying attention that even though it converges to another local maxima and leads to a different distribution, the means for the mixture of Gaussian are still close to the

true peaks of our testing distribution. In Section **4.2** (4.17) we provide an approach to get an estimate of the normalization constant. In practice this estimate usually gives moderate values and thus leads to good fitting results.



(a) original (b) multiply by 5 (c) multiply by 12

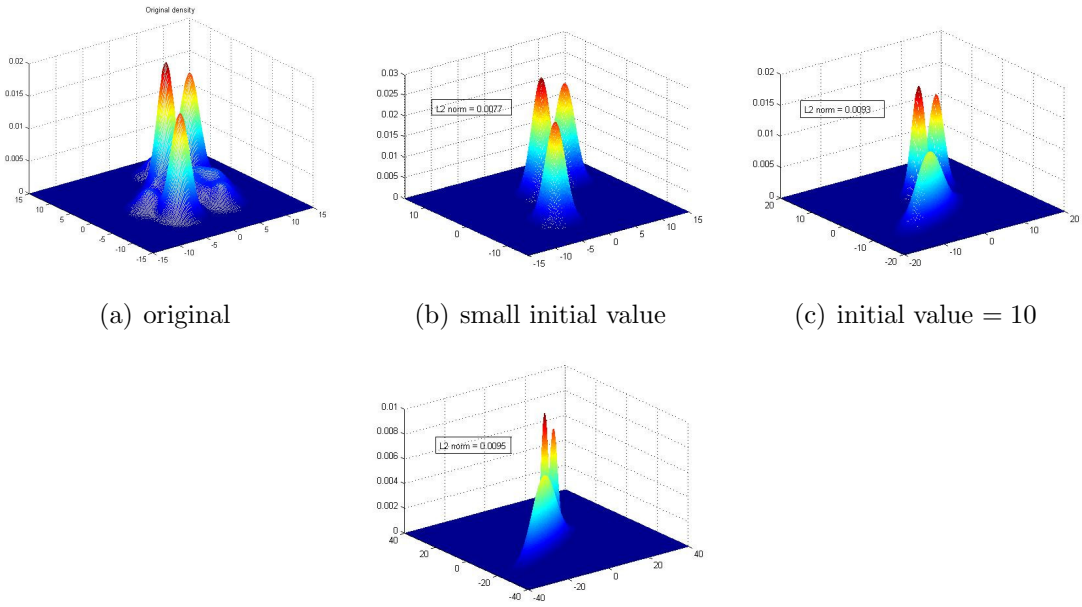
Figure 4.9: **Mixture of Gaussian fitting with density subject to a constant.** Figure (a) is the original density which contains three big modes; Figure (b) and (c) are the fitting result for normalization constant 5 and 12.

Experiment 7: Hidden Markov Model Experiments 1-6 have illustrated the performance of the clustering method based on the L_p measurement but only in the cases of two dimensional distributions. In fact, as we emphasized in the introduction section, this method could also be applied to high dimensional spaces. However, to demonstrate the ability for the spaces of more than two dimensions is difficult due to the lack of plot illustrations. In this experiment, we perform the following hidden Markov experiment, in which we could intuitively figure out the answer using our Minimum L_p Norm Estimate approach.

The HMM model: $x_1, x_2, ..., x_n$ are the hidden variables with two possible states, 0 and 1, and with transition probability

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}$$

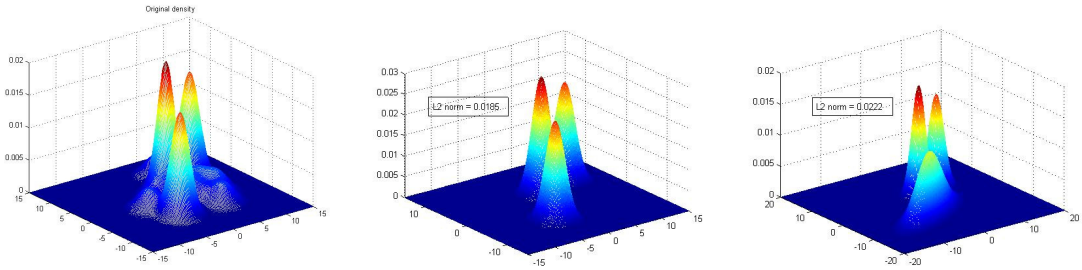
$y_1, y_2, ..., y_n$ are the observable random variables and $y_i \sim N(x_i, 1)$.



(a) original (b) small initial value (c) initial value = 10

(d) initial value = 20

Figure 4.10: **Effect of initial values for normalization constant.** Figure (a) is the original density which contains three big modes; After multiply a constant 5,initial values of 0.5, 1, 2,3,5,7 all converge to this distribution with normalization constant 3.2824; Large initial value of 10,20 converge to the normalization constant 5.2449. Figure (b)-(d) gives corresponding fitted distributions.



(a) original (b) original (c) original

Figure 4.11: **Effect of initial values for normalization constant.** Figure (a) is the original density which contains three big modes; After multiple a constant 12, initial values of 2,4,8,12,15,20 all converge to the distribution in (b) with normalization constant 7.8778; Large initial value of 25 converge to (c) with normalization constant 12.2449

We are interested in the distribution of the hidden variables $x_1, x_2, \cdots, x_n$. Notice that for the hidden Markov model, actually we are able to calculate the probabilities exactly using dynamic programming. In this experiment, we perform the Minimum L_p Norm idea to look for the clustering characteristics and compare the results. Let us consider the specific example with initial probability $p_0 = (1, 0)$ and $n = 4$, i.e. this hidden Markov model always starts from state 0. Then, we fit this distribution with mixture of two four-dimensional Gaussian distributions. After 100 iterations, we obtain the clusters (two means of Gaussian)

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 0.4765 & 0.8883 & 1.0741 & 1.0933 \\ 0.0300 & 0.0592 & 0.1253 & 0.2334 \end{bmatrix}$$

with mixing probabilities 0.3057 and 0.6943. However, does this result make sense? Let us compare this answer with the following thought experiment.

Thought experiment: The highest five probabilities of hidden states are as follows:

$$\begin{aligned} \text{Pro}(x_4^1 = (0,0,0,0)) &= 0.4096 \\ \text{Pro}(x_4^1 = (1,1,1,1)) &= 0.1458 \\ \text{Pro}(x_4^1 = (0,1,1,1)) &= 0.1296 \\ \text{Pro}(x_4^1 = (0,0,1,1)) &= 0.1152 \\ \text{Pro}(x_4^1 = (0,0,0,1)) &= 0.1024 \end{aligned}$$

We find that $(0, 0, 0, 0)$ is kind of isolated from other masses, but it gets the probability of 0.4. Thus, we may guess one of the two means might be selected by the following average:

$$\begin{aligned} \tilde{\mu_1} =& (0,0,0,0) + \epsilon\Big(\frac{0.1458}{\sqrt{4}}(1,1,1,1) + \frac{0.1296}{\sqrt{3}}(0,1,1,1) + \frac{0.1152}{\sqrt{2}}(0,0,1,1) \\ &+ \frac{0.1024}{\sqrt{1}}(0,0,0,1)\Big), \end{aligned} \tag{4.18}$$

where ϵ is a small positive number. Then the rest of masses would then concentrate around the following average:

$$\tilde{\mu_2} = \frac{0.1458}{0.4957}(1,1,1,1) + \frac{0.1296}{0.4957}(0,1,1,1) + \frac{0.1152}{0.4957}(0,0,1,1) + \frac{0.1024}{0.4957}(0,0,0,1),$$

where $0.4957 = 0.1458 + 0.1296 + 0.1152 + 0.1024$. The corresponding mixing probabilities are about $0.4 - \tilde{\epsilon}$ and $0.6 + \tilde{\epsilon}$ with some small positive $\tilde{\epsilon}$. Hence, the result is very much like what we would imagine above.

4.4 Discussion and Future Directions

In this chapter we propose the Minimum L_p Norm Estimator to reveal the clustering characteristic of underlying probability spaces through mixture model. This new estimator is different from the standard approach in that in addition to the sampled values, it also take use of the exact probabilities (or densities, or subject to a normalization constant). We show that with different choice of p, the minimum L_p estimator might be able to reveal different characteristics. In addition, during the derivation of this estimator, we don't need to assign clusters for each sampled values, which avoids the potential error involved with assigning clusters.

Evaluating expectation is another big problem which is usually not computationally feasible to compute exactly (e.g. for discrete probability distribution with enormous states). Very typically, people use Monte Carlo method to approximate the expectation $E\Big[f(x)\Big] \sim \frac{1}{N}\sum_{i=1}^{N} f(x_i)$. However, the same situation arises: we not only have samples, "x_i" but also have the probabilities of the samples, "$p(x_i)$" which are unused in the original Monte Carlo approximation. With the probabilities of the samples, "$p(x_i)$", can we obtain better approximation of the expectation than Monte Carlo method?

Unlike discrete distributions, the expectations under continuous distribution have been widely studied using both samples and their probabilities, e.g. importance sampling. Importance sampling is a variance reduction technique that can be used in the

Monte Carlo method. The idea behind importance sampling is that certain values of the input random variables in a simulation have more impact on the parameter being estimated than others. If these "important" values are emphasized by sampling more frequently, then the estimator variance can be reduced. Hence, the basic methodology in importance sampling is to choose a distribution which "encourages" the important values. This idea is basically the same as taking use of the information of density term and it seems very natural that we would like to put more weight on the ensembles with higher density since they should have more contribution to the expectation.

For discrete cases, in Chapter 3 we already discussed some situations where we would have at least half probability to get better estimate of the expectation with the density information than the Monte Carlo methods. Let us recall the Theorem here.

Theorem 3.2.3: *Suppose the discrete high-dimensional space has large number of ensembles and satisfies the condition that for every point value x_i, $P(x_i) << \frac{1}{N}$ for the sample size N. We also assume that the N samples are different to each other, i.e. each with empirical frequency $\frac{1}{N}$. Then when $J(.)$ is a monotonic continuous function, $\hat{G}$ has at least half probability to behave better than $\tilde{G}$. i.e.*

$$P(|\tilde{G} - E[G(X)]| \geq |\hat{G} - E[G(X)]|) \geq \frac{1}{2}$$

As a direct application, Theorem **3.2.3** could be applied to the estimate of entropy $H(X)$ for high dimensional spaces since $H(X) = E\left[\frac{1}{p(x)}\right]$ and the function $\frac{1}{x}$ is monotone. The high dimensional spaces should satisfy the condition that the probability of each ensemble is pretty small, e.g. the posterior space of RNA secondary structure might be such a situation.

This is just a very simple example which we showed to be able to benefit from the extra information $p(x)$. We believe there is still a lot of space to fill in this problem. Ideally we hope to find a general estimator involving both samples and

their probabilities or densities, which would give better estimate for expectation than the Monte Carlo estimate.

Part IV

Conclusion and Future direction

In this thesis, we first apply hidden Markov model to paleoclimatology study. Chapter 1 gives a direct application of two state hidden Markov model. Chapter 2 focuses on building a pairwise hidden Markov model based on the deterministic dynamic programming algorithm. The goal is to implement the alignment between pairs of paleoclimate records and provide uncertainty analysis. In addition to Viterbi alignment, i.e. the most probable alignment, Centroid alignment is also investigated. Centroid alignment has better behavior for credibility limits, i.e. it has tighter credibility bands and always lie between the upper bound and low bound since Centroid alignment is actually equivalent to the median alignment. There still exist several possible extensions, which might add to the power of the current model. In addition to pairwise alignment, we would extend the current model into multiple alignment situations, i.e. profile HMM approach. Furthermore, instead of dividing the time scale into several hundred subintervals and implementing mappings via these subintervals, we will explore continuous time-step approach, which is expected to be more generally applicable, and continue to investigate its versatility. Also, other alternative approach to approximate the values for the between points instead of linear interpolation would also be explored.

Computational biology has recently been a fruitful source of high-dimensional discrete inference problems. In Chapter 3 we present and discuss the characterization of high dimensional space by informative variables. As an application, we study the shape of the posterior space of RNA secondary structure. In addition to exact calculation, we provide sampling based algorithms for the inference and optimization procedure, along with a principled approach to parameter estimation. Another promising application is the importance of informative base pairs in the prediction of RNA secondary structure. There are many problems in computational biology that would benefit from informative base pairs, like alternative splicing, simultaneous alignment, structure prediction of RNA and genome rearrangement. These problems represent attractive and promising directions for future applied work. Furthermore, the reduced model might also be explored due to the fact that there exist a large amount of base pairs which have entropy close to zero and thus are non-

informative to the space. We argue that the reduced model should be similar to the complete model, while the reduced model reduces the dimensionality to a large degree.

In Chapter 4 we propose a new approach to look for modes of a distribution using not only samples but also their probabilities or densities. We propose the minimum empirical L_p norm estimator to fit the distribution with mixture model where in the experiment we use mixture of Gaussian. We also show that this estimator has consistency property and investigate the limit phenomena of p. Our new estimator reveals different perspective from maximum likelihood estimator which seems more sensitive to tails, while our estimator is more stable and robust in locating the main modes. We demonstrate the abilities of this approach for high dimensional space by experimenting on the hidden Markov model with thought experiment comparison.

There are many applications to be explored using this method. For example, it can be applied to the identification problems of hidden states in pattern recognition applications. In particular, in image recognition and detection problems, implementing this method on the posterior probability distribution provides us the clusters representing the hidden poses of objects. Moreover, we can characterize different features of the distribution besides looking for modes. As we mention in Chapter 4, the mixture model $f_\theta(x)$ could be any parametric form which would help reveal different features of the distribution.

In addition to clustering problems, estimating expectation is another important issue for computing or characterizing high dimensional data. People encounter huge amount of applications of x-px problems in this matter. Those problems also relate to importance sampling problems and involve the issues of computability of normalization constants. The future challenge will be to answer the following questions: can we have an estimator involving samples, X_i's and their probabilities $P(X_i)$'s (or up to a normalization constant) better than traditional estimators like Monte Carlo estimator? Or in which scenarios we can do better?

Bibliography

[1] R.A. Abagyan, *Towards protein folding by global energy optimization*, FEBS Lett **325** (1993), 17–22.

[2] Mark Altabet, *Isotopic tracers of the marine nitrogen cycle: Present and past*, Springer Berlin Heidelberg, 2006.

[3] C.B. Anfinsen, *Principles that govern the folding of protein chains*, Science **181** (1973), 223–230.

[4] J. Baker, *The dragon system–an overview*, Acoustics, Speech and Signal Processing, IEEE Transactions on **23** (1975), 24–29.

[5] J.K. Baker, *The dragon systeman overview*, IEEE Trans. Acoust. Speech Signal Processing **ASSP-23** (1975), 24–29.

[6] R. Bakis, *Continuous speech recognition via centisecond acoustic states*, Acoustical Society of America **59** (1976), S97–S97.

[7] Richard T. Barber and Francisco P. Chavez, *Biological consequences of el nio*, Science **222** (1983), no. 4629, 1203–1210.

[8] Alex Bateman, Ewan Birney, Lorenzo Cerruti, Richard Durbin, Laurence Etwiller, SeanR. Eddy, Sam Griffiths-Jones, Kevin L. Howe, Mhairi Marshall, and Erik L. L. Sonnhammer, *The pfam protein families database*, Nucleic Acids Research **30** (2002), no. 1, 276–280.

[9] LE Baum, *An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes*, Inequalities **3** (1972), 1–8.

[10] L.E. Baum and J. A. Egon, *An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology*, Bull. Amer. Meteorol. Soc. **73** (1967), 360–363.

[11] L.E. Baum and T. Petrie, *Statistical inference for probabilistic functions of finite state markov chains*, Ann. Math. Stat **37** (1996), 1554–1563.

[12] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains*, Ann. Math. Stat. **41** (1970), 164–171.

[13] D.M. Blei, A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation*, Journal of Machine Learning Research **3** (2003), 993–1022.

[14] L. E. Carvalho and C. E. Lawrence, *Centroid estimation in discrete high-dimensional spaces with applications in biology*, PNAS **105** (2008), 3209–3214.

[15] Luis E. Carvalho, *Bayesian centroid estimation*, Ph.D. thesis, Brown University, 2009.

[16] Morales C.E., Hormazabal S.E., and Blanco J., *Interannual variability in the mesoscale distribution of the depth of the upper boundary of the oxygen minimum layer off northern chile (18-24s): Implications for the pelagic system and biogeochemical cycling*, Journal of Marine Research **57** (1999), 909–932.

[17] Chi Yu Chan, Charles E. Lawrence, and Ye Ding, *Structure clustering features on the sfold web server*, Bioinformatics **21**, no. 20, 3926–3928.

[18] C.Y. Chan, C E. Lawrence, and Y. Ding, *Structure clustering features on the sfold web server*, Bioinformatics **21** (2005), 3926–3928.

[19] Caitlin R. Chazen, Mark A. Altabet, and Timothy D. Herbert, *Abrupt mid-holocene onset of centennial-scale climate variability on the peru-chile margin*, Geophys. Res. Lett. **36** (2009), no. 18, L18704–.

[20] A. C. Clement, R. Seager, and M. A. Cane, *Orbital controls on the el nio/southern oscillation and the tropical climate*, Paleoceanography **14** (1999), no. 4, 441–456.

[21] L.A Codispoti and J.P Christensen, *Nitrification, denitrification and nitrous oxide cycling in the eastern tropical south pacific ocean*, Marine Chemistry **16** (1985), no. 4, 277 – 300, ¡ce:title¿Aquatic Nitrogen Cycles-a Session convened at the joint meeting of the American Geophysical Union and the American Society for Limnology and Oceanography¡/ce:title¿.

[22] Jessica L. Conroy, Jonathan T. Overpeck, Julia E. Cole, Timothy M. Shanahan, and Miriam Steinitz-Kannan, *Holocene changes in eastern tropical pacific climate inferred from a galpagos lake sediment record*, Quaternary Science Reviews **27** (2008), no. 1112, 1166 – 1180.

[23] Rena Czeschel, Lothar Stramma, Franziska U. Schwarzkopf, Benjamin S. Giese, Andreas Funk, and Johannes Karstensen, *Middepth circulation of the eastern tropical south pacific and its link to the oxygen minimum zone*, J. Geophys. Res. **116** (2011), no. C1, C01015–.

[24] S. Datta and S. Datta, *Comparisons and validation of statistical clustering techniques for microarray gene expression data:*, Bioinformatics **19** (2003), 459–466.

[25] Curtis Deutsch, Holger Brix, Taka Ito, Hartmut Frenzel, and LuAnne Thompson, *Climate-forced variability of ocean hypoxia*, Science **333** (2011), no. 6040, 336–339.

[26] Y. Ding, *Rational statistical design of antisense oligonucleotides for high throughput functional genomics and drgu target validation*, Statistica Sinica **12** (2002), 273–296.

[27] Y. Ding, C.Y. Chan, and C.E. Lawrence, *Rna secondary structure prediction by centroids in a boltzmann weighted ensemble*, RNA **11** (2005), 1157–1166.

[28] Y. Ding and C.E. Lawrence, *Statistical prediction of single-stranded regions in rna secondary structure and application to predicting effective antisense target sites and beyond*, Necleic Acids Res **29** (2001), 1034–1046.

[29] Ye Ding, Chi Yu Chan, and Charles E. Lawrence, *Clustering of rna secondary structures with application to messenger rnas*, Journal of Molecular Biology **359** (2006), no. 3, 554 – 571.

[30] Ye Ding and Charles E. Lawrence, *A statistical sampling algorithm for rna secondary structure prediction*, Nucleic Acids Research **31** (2003), no. 24, 7280–7301.

[31] S. R. Eddy and R. Durbin, *Rna sequence analysis using covariance models*, Nucleic Acid Research **22** (1994), 2079–2088.

[32] C Ehresmann, F Baudin, M Mougel, P Romby, J P Ebel, and B Ehresmann, *Probing the structure of rnas in solution*, Nucleic Acids Res **15** (1987), 9109–9128.

[33] Christine Euler and Ulysses S. Ninnemann, *Climate and antarctic intermediate water coupling during the late holocene*, Geology **38** (2010), no. 7, 647–650.

[34] M.A.T. Figueiredo and A.K. Jain, *Unsupervised learning of finite mixture models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002), 381–396.

[35] Daifang Gu and S G. H. Philander, *Interdecadal climate fluctuations that depend on exchanges between the tropics and extratropics*, Science **275** (1997), no. 5301, 805–807.

[36] James M. Hart, Scott D. Kennedy, David H. Mathews, and Douglas H. Turner, *Nmr-assisted prediction of rna secondary structure: Identification of a proba-*

ble pseudoknot in the coding region of an r2 retrotransposon, Journal of the American Chemical Society **130** (2008), no. 31, 10233–10239.

[37] Gerald H. Haug, Konrad A. Hughen, Daniel M. Sigman, Larry C. Peterson, and Ursula Rhl, *Southward migration of the intertropical convergence zone through the holocene*, Science **293** (2001), no. 5533, 1304–1308.

[38] John J. Helly and Lisa A. Levin, *Global distribution of naturally occurring marine hypoxia on continental margins*, Deep Sea Research Part I: Oceanographic Research Papers **51** (2004), no. 9, 1159–1168.

[39] I.L. Hofacker, *Vienna rna secondary structure server*, Nucleic Acids Res **31** (2003), 3429–3431.

[40] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M. Tacker, and P. Schuster, *Fast folding and comparisons of rna secondary structure*, Monatshefte f. Chemie **125** (1994), 167–188.

[41] Rui Xin Huang, *The generalized eastern boundary conditions and the three-dimensional structure of the ideal fluid thermocline*, J. Geophys. Res. **94** (1989), no. C4, 4855–4865.

[42] Peter Huybers and Carl Wunsch, *A depth-derived pleistocene age model: Uncertainty estimates, sedimentation variability, and nonlinear climate change*, Paleoceanography **19** (2004), no. 1, PA1028–.

[43] A. Huyer, A., R.L. Smith, and T. Paluszkiewicz, *Coastal upwelling off peru during normal and el nino times*, J. Geophys. Res. **92 (C13)** (1987), 14297–307.

[44] H. Stommel J. R. Luyten, J. Pedlosky, *The ventilated thermocline*, J. Phys. Oceanogr **13** (1983), 292–309.

[45] F. Jelinek, *Continuous speech recognition by statistical methods*, Proceedings of the IEEE **64** (1976), 532–556.

[46] Johannes Karstensen, *Formation of the south pacific shallow salinity minimum: A southern ocean pathway to the tropical pacific*, J. Phys. Oceanogr. **34** (2004), no. 11, 2398–2412.

[47] Johannes Karstensen, Lothar Stramma, and Martin Visbeck, *Oxygen minimum zones in the eastern tropical atlantic and pacific oceans*, Progress In Oceanography **77** (2008), no. 4, 331 – 350, ¡ce:title¿A New View of Water Masses After WOCE. A Special Edition for Professor Matthias Tomczak¡/ce:title¿.

[48] L. Kaufman and P.J. Rousseuw, *Finding groups in data: An introduction to cluster analysis*, John Wiley & Sons, New York (1990).

[49] Athanasios Koutavas, Peter B. deMenocal, George C. Olive, and Jean Lynch-Stieglitz, *Mid-holocene el ni aisouthern oscillation (enso) attenuation revealed by individual foraminifera in eastern tropical pacific sediments*, Geology **34** (December, 2006), no. 12, 993–996.

[50] Athanasios Koutavas, Jean Lynch-Stieglitz, Thomas M. Marchitto, and Julian P. Sachs, *El nino-like pattern in ice age tropical pacific sea surface temperature*, Science **297** (2002), no. 5579, 226–230.

[51] Hans Kunsch, Stuart Geman, and Athanasios Kehagias, *Hidden markov random fields*, Annals of Applied Probability, **5** (1995), 577–602.

[52] S. Levitus and T. P. Boyer, *World ocean atlas: 1994, vol. 4, temperature*, Natl. Environ. Satell., Data, and Inf. Serv., NOAA,Washington, D. C. (1994).

[53] S.M. Libes and W.G. Deuser, *The isotope geochemistry of particulate nitrogen in the peru upwelling area and the gulf of maine*, Deep Sea Research Part A. Oceanographic Research Papers **35** (1988), no. 4, 517 – 533.

[54] F. Lipschultz, S.C. Wofsy, B.B. Ward, L.A. Codispoti, G. Friedrich, and J.W. Elkins, *Bacterial transformations of inorganic nitrogen in the oxygen-deficient waters of the eastern tropical south pacific ocean*, Deep Sea Research Part A. Oceanographic Research Papers **37** (1990), no. 10, 1513 – 1541.

[55] Lorraine E. Lisiecki and Philip A. Lisiecki, *Application of dynamic programming to the correlation of paleoclimate records*, Paleoceanography **17** (2002), no. 4, 1049–.

[56] Lorraine E Lisiecki and Maureen E Raymo, *Pliocene-Pleistocene stack of globally distributed benthic stable oxygen isotope records*, 2005.

[57] J. B. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability (L. M. Le Cam and J. Neyman, eds.), vol. 1, University of California Press, 1967, pp. 281–297.

[58] Matthew C. Makou, Timothy I. Eglinton, Delia W. Oppo, and Konrad A. Hughen, *Postglacial changes in el nino and la nina behavior*, Geology **38** (2010), no. 1, 43–46.

[59] Douglas G. Martinson, William Menke, and Paul Stoffa, *An inverse approach to signal correlation*, J. Geophys. Res. **87** (1982), no. B6, 4807–4818.

[60] Douglas G Martinson, Nicklas G Pisias, James D Hays, John D Imbrie, Theodore C Moore, and Nicholas J Shackleton, *Age dating and isotopic analyses of sediment core RC11-120*, 1987, Supplement to: Martinson, Douglas G; Pisias, Nicklas G; Hays, James D; Imbrie, John D; Moore, Theodore C; Shackleton, Nicholas J (1987): Age Dating and the orbital theory of the ice ages: development of a high-resolution 0 to 300,000-year chronostratigraphy. Quaternary Research, 27, 1-29, doi:10.1016/0033-5894(87)90046-9.

[61] David H Mathews and Douglas H Turner, *Prediction of rna secondary structure by free energy minimization*, Current Opinion in Structural Biology **16** (2006), no. 3, 270 – 278, ¡ce:title¿Nucleic acids/Sequences and topology¡/ce:title¿ ¡ce:subtitle¿Anna Marie Pyle and Jonathan Widom/Nick V Grishin and Sarah A Teichmann¡/ce:subtitle¿.

[62] J.S. McCaskill, *The equilibrium partition function and base pair binding probabilities for rna secondary structure*, Biopolymers **29** (1990), 1105–1119.

[63] G.L. McLachlan and D. Pell, *Finite mixture models*, Wiley, 2000.

[64] Christopher M. Moy, Geoffrey O. Seltzer, Donald T. Rodbell, and David M. Anderson, *Variability of el nino southern oscillation activity at millennial timescales during the holocene epoch*, Nature **420** (2002), no. 6912, 162–165.

[65] Saul B. Needleman and Christian D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequence of two proteins*, Journal of Molecular Biology **48** (1970), no. 3, 443 – 453.

[66] Peter R. Oke and Matthew H. England, *Oceanic response to changes in the latitude of the southern hemisphere subpolar westerly winds*, J. Climate **17** (2004), no. 5, 1040–1054.

[67] R.V. Pappu, G.R. Marshall, and J.W. Ponder, *A potential smoothing algorithm accurately predicts transmembrane helix packing*, Nat. Struct. Biol **6** (1999), 50–55.

[68] H Peng, Fuhui Long, and Chris Ding, *Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005), 1226–1238.

[69] Warren L. Prell, John Imbrie, Douglas G. Martinson, Joseph J. Morley, Nicklas G. Pisias, Nicholas J. Shackleton, and Harold F. Streeter, *Graphic correlation of oxygen isotope stratigraphy application to the late quaternary*, Paleoceanography **1** (1986), no. 2, 137–162.

[70] Tangdong Qu, Shan Gao, Ichiro Fukumori, Rana A. Fine, and Eric J. Lindstrom, *Subduction of south pacific waters*, Geophys. Res. Lett. **35** (2008), no. 2, L02610–
.

[71] Elizabeth E. Regulski and Ronald R. Breaker, *In-line probing analysis of riboswitches*, Methods in Molecular Medicine **419** (2008), 53–67.

[72] Donald T. Rodbell, Geoffrey O. Seltzer, David M. Anderson, Mark B. Abbott, David B. Enfield, and Jeremy H. Newman, *An 15,000-year record of el nino-driven alluviation in southwestern ecuador*, Science **283** (1999), no. 5401, 516–520.

[73] H. Sakoe and S. Chiba, *Dynamic programming algorithm optimization for spoken word recognition*, Acoustics, Speech and Signal Processing, IEEE Transactions on **26** (1978), 43–49.

[74] Wolfgang Schneider, Rosalino Fuenzalida, Efraln Rodrlguez-Rubio, JosE GarcEs-Vargas, and Luis Bravo, *Characteristics and formation of eastern south pacific intermediate water*, Geophys. Res. Lett. **30** (2003), no. 11, 1581–.

[75] Nicholas J Shackleton, Michael A Hall, and D Pate, *Pliocene stable oxygen and carbon isotope record of benthic foraminifera from ODP Site 138-846 in the eastern equatorial Pacific*, 1995.

[76] Bruce A Shapiro, Yaroslava G Yingling, Wojciech Kasprzak, and Eckart Bindewald, *Bridging the gap in rna structure prediction*, Current Opinion in Structural Biology **17** (2007), no. 2, 157 – 165, ¡ce:title¿Theory and simulation / Macromolecular assemblages¡/ce:title¿.

[77] A. E. Shevenell, A. E. Ingalls, E. W. Domack, and C. Kelly, *Holocene southern ocean surface temperature variability west of the antarctic peninsula*, Nature **470** (2011), no. 7333, 250–254.

[78] T.F. Smith and M.S. Waterman, *Identification of common molecular subsequences*, Journal of Molecular Biology **147** (1981), no. 1, 195 – 197.

[79] Lowell Stott, Christopher Poulsen, Steve Lund, and Robert Thunell, *Super enso and global climate oscillations at millennial time scales*, Science **297** (2002), no. 5579, 222–226.

[80] J. R. Toggweiler, K. Dixon, and W. S. Broecker, *The peru upwelling and the ventilation of the south pacific thermocline*, J. Geophys. Res. **96** (1991), no. C11, 20467–20497.

[81] TK Vintsjuk, *Recognition of words of oral speech by dynamic programming*, (1968).

[82] Bobbie-Jo M. Webb-Robertson, Lee Ann McCue, and Charles E. Lawrence, *Measuring global credibility with application to local sequence alignment*, PLoS Comput Biol **4** (2008), no. 5, e1000077–.

[83] Kevin A. Wilkinson, Edward J. Merino, and Kevin M. Weeks, *Rna shape chemistry reveals nonhierarchical interactions dominate equilibrium structural transitions in trnaasp transcripts*, Journal of the American Chemical Society **127** (2005), no. 13, 4659–4667.

[84] J Zhu, J S Liu, and C E Lawrence, *Bayesian adaptive sequence alignment algorithms.*, Bioinformatics **14** (1998), no. 1, 25–39.

[85] M. Zuker, *On finding all suboptimal foldings of an rna molecule*, Science **244** (1989), 48–52.