Modeling Temporal and Spatial Data Dependence with Bayesian Nonparametrics

by

Lu Ren

Department of Electrical and Computer Engineering
Duke University

Date: ______________________

Approved:

____
Lawrence Carin, Advisor

____
Jeffrey Krolik

____
Loren Nolte

____
David Dunson

____
Mauro Maggioni

____
Scott Lindroth

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2010

Abstract
(EE)

Modeling Temporal and Spatial Data Dependence with Bayesian Nonparametrics

by

Lu Ren

Department of Electrical and Computer Engineering
Duke University

Date: ______________________

Approved:

____
Lawrence Carin, Advisor

____
Jeffrey Krolik

____
Loren Nolte

____
David Dunson

____
Mauro Maggioni

____
Scott Lindroth

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2010

Abstract

In this thesis, temporal and spatial dependence are considered within nonparametric priors to help infer patterns, clusters or segments in data. In traditional nonparametric mixture models, observations are usually assumed exchangeable, even though dependence often exists associated with the space or time at which data are generated. Focused on model-based clustering and segmentation, this thesis addresses the issue in different ways, for temporal and spatial dependence.

For sequential data analysis, the dynamic hierarchical Dirichlet process is proposed to capture the temporal dependence across different groups. The data collected at any time point are represented via a mixture associated with an appropriate underlying model; the statistical properties of data collected at consecutive time points are linked via a random parameter that controls their probabilistic similarity. The new model favors a smooth evolutionary clustering while allowing innovative patterns to be inferred. Experimental analysis is performed on music, and may also be employed on text data for learning topics.

Spatially dependent data is more challenging to model due to its spatially-grid structure and often large computational cost of analysis. As a non-parametric clustering prior, the logistic stick-breaking process introduced here imposes the belief that proximate data are more likely to be clustered together. Multiple logistic regression functions generate a set of sticks with each dominating a spatially localized segment. The proposed model is employed on image segmentation and speaker diarization, yielding generally homogeneous segments with sharp boundaries.

In addition, we also consider a multi-task learning with each task associated with spatial dependence. For the specific application of co-segmentation with multiple images, a hierarchical Bayesian model called H-LSBP is proposed. By sharing the same mixture atoms for different images, the model infers the inter-similarity between each pair of images, and hence can be employed for image sorting.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

BIC	Bayesian information criterion.
AIC	Akaike's information criterion.
DP	Dirichlet process.
CPR	Chinese restaurant process.
DDP	dependent Dirichlet process.
GSDP	general spatial Dirichlet process.
LDA	latent Dirichlet allocatoin.
RCRP	recurrent Chinese restaurant process.
KSBP	kernel stick-breaking process.
HDP	Hierarchical Dirichlet process.
nDP	nested Dirichlet process.
HMM	hidden Markov model.
iHMM	infinite hidden Markov model.
dHDP	dynamic hierarchical Dirichlet process.
LSBP	logistic stick-breaking process.
H-KSBP	hierarchical kernel stick-breaking process.
H-LSBP	hierarchical logistic stick-breaking process.
MCMC	Markov chain Monte Carlo.
iGMM	infinite Gaussian mixture model.
DMDP	dynamic mixture Dirichlet process.

MFCCs	Mel frequency cepstral coefficients.
VQ	vector quantization.
VB	variational Bayesian.
GP	Gaussian process.
MRF	Markov random field.
Ncuts	normalized cuts.
Stu.-t MM	student-t distribution mixture model
RI	Rand Index.
VoI	Variation of Information.
SDP	spatial Dirichlet process

Acknowledgements

First of all, I would like to express my heartfelt gratitude to my advisor, Dr. Lawrence Carin, for his guidance, inspiration, encouragement and patience throughout my graduate research work. Without his tremendous support, this work would not be possibly finished. His influence on my thought and working attitude will continue beyond the completion of this thesis.

I am grateful to Dr. David Dunson, for his insightful suggestions and valuable comments on my research. Special thanks to Dr. Scott Lindroth for his great support on the theoretical analysis for the music. I also need to particularly thank the other committee members: Dr. Jeffrey Krolik, Dr. Loren Nolte and Dr. Mauro Maggioni, for their time and advice on this thesis.

I would like to thank all my research group members: Dr. Xuejun Liao, Dr. Balaji Krishnapuram, Dr. Ya Xue, Dr. Shihao Ji, Dr. Hui Li, Dr. Qiuhua Liu, Dr. Lihan He, Dr. Yuting Qi, Dr. Dehong Liu, Dr. Kai Ni, Dr. Qi An, Dr. Iulian Pruteanu, Dr. Lan Du, Chunping Wang, John Paisley, Minhua Chen, Eric Wang and Haojun Chen for their collaboration, assistance and valuable discussions. I really enjoy the time spent with them at Duke and have benefited a lot from such an experience.

I especially thank my parents for their encouragement and support during these five years. Their love means a lot to me and will accompany me forever. Finally, I am very thankful to my husband and my daughter to be born. Thanks to their love and support!

1

Introduction

Machine learning is a process of exploring the collected observations from perceptions and generating a learning rule for prediction of the incoming data. The "rule" may be represented in terms of a mapping function, translating the encoded information from predictors to a classifier; or it may be a generative model, approximating the underlying distribution of the observed data. For supervised learning, labels are provided for collected samples and treated as inputs during the "training " process. However, sometimes it is surprisingly costly to label observations. For example, recording a video is virtually free, but it is very expensive and time-consuming to annotate the scene and topic for each frame.

The most popular solution to the "unlabeled" problem is to cluster observations into groups via a mixture of density functions. This unsupervised model learns unknown world depending on the data itself, and can also be treated as a "data preprocessing" step to help acquire the labels easily. For example, unsupervised learning is employed for speaker diarization. Provided a spoken document consisting of multiple speakers, speaker diarization is segmenting the audio signal into contiguous temporal regions, and within a given region a particular individual is speaking.

For speaker diarization, three problems need to be solved: (i) How many speakers participated in the talk? (ii) When did a speaker begin his/her talk? (iii) How long did each speaking continue?

This is a typical multimodal-learning task. We can cluster the data with a Gaussian mixture model, with each Gaussian parameter representing one speaker's pattern. To consider the first problem, we might fix the number of Gaussian components as a constant or do model selection with cross-validation. We alternatively infer the number of model parameters in a nonparametric manner, and the model complexity is allowed to adaptively change as the data need. For the second problem, the model performance depends on the identifiability of each inferred Gaussian parameter. These parameters inferred from the data generally represent various utterances, and need to be robust to the vocal instability. The third problem of learning duration is equivalent to finding the hidden state underlying each observation. For a temporally smooth constraint, a Markovian dependence is included within the model assumption.

Motivated by the three challenging problems, we are interested in a category of nonparametric probabilistic models, allowing a temporal or spatial dependence learned from the posterior. Although the study is employed on several unsupervised learning applications here, it may be extended to other supervised or semi-supervised learning problems in the future.

1.1 Nonparametric Probabilistic Models

For data density estimation, a generative model is usually assumed underlying the observations. Each observation $\mathbf{x}$ is i.i.d. drawn from certain distribution $F(\theta)$, with the unknown distribution function $F(\cdot)$ and associated parameter θ. For example, we can guess that $F(\theta)$ is a multivariate Gaussian distribution. Then the parameter

θ represents both the mean and covariance matrix. Sometimes a unimodal model is too rigid to describe a complicated data density, such as Figure 1.1 with five Gaussian components included. Therefore, a mixture model is considered for such a case.

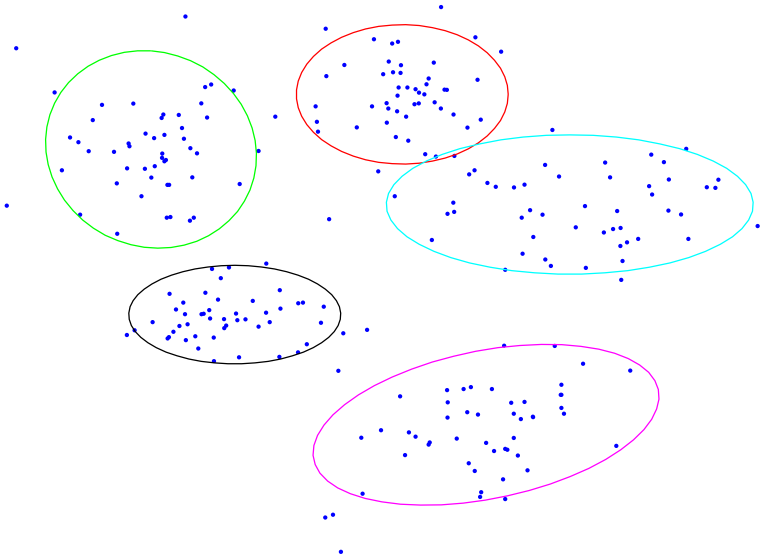


FIGURE 1.1: A typical Gaussian Mixture.

The density function for a finite mixture with K components is

$$f(x) = \sum_{k=1}^{K} \pi_k f(x|\theta_k), \quad \text{with} \quad 0 < \pi_k < 1, \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1, \tag{1.1}$$

in which π_k is the prior weight for data x being drawn from the component k, and $f(x|\theta_k)$ represents the likelihood function of x given the parameter θ_k. Given the true value of K and sufficient data samples, it usually yields a good estimate of the mixture weights and model parameters; however, K is unknown for most cases in advance, and an inappropriate guess of K might degrade the validation of the model inference, leading to over- or under-fitting issues. Some parametric approaches based on the likelihood are usually considered for model selection, such as hypothesis testing [TK03], cross validation [Koh95], Bayesian information criterion (BIC) [Sch78], Akaike's information criterion (AIC) [Aka74]; or a Poisson prior is chosen for K within the model and a posterior estimate can be achieved. We here address this challenge using nonparametric statistical models.

"Nonparametric" does not mean that no parameters exist in the model. It just indicates that the number of parameters may change as new data are observed. A flexible nonparametric prior is expected to model any arbitrary distribution and can be defined as a measure for infinite distributions. The Dirichlet process (DP) [BM73], studied in Bayesian data analysis, allows such a possibility. It can be treated as a prior for an infinite mixture probabilistic model or directly used to model the data density.

A Dirichlet process (DP) is parameterized with a base measure G_0 and a positive scaling parameter α. The DP mixture model generates data $\{x_i\}_{i=1}^{N}$ as follows:

$$G \sim DP(\alpha G_0) \quad \theta_i \overset{i.i.d.}{\sim} G \quad x_i \sim F(\theta_i) \tag{1.2}$$

G_0 can be any continuous or discrete distribution for the atom locations (see Figure 1.2), and it is the expectation of G; α controls the variance of G from G_0. For each data x_i, the associated parameter θ_i is i.i.d. drawn from G, a discrete distribution drawn from $DP(\alpha G_0)$. A cluster is defined as the group of data sharing the same parameter, and the expected number of clusters given the current observations is $\alpha \log N$.

There are many ways to look at the Dirichlet process. One of them is the Chinese restaurant process (CRP) [Ald85], a random process with N customers assigned to infinite number of tables. One table represents one cluster and the number of customers sitting at each table is uncertain at the beginning. As a new customer comes in, he can either choose one of the old tables that shared by the previous ones, or just take a new table by himself. All the customers sitting at the same table share the same dish. Figure 1.3 illustrates one example of the random process, in which $\{\phi_1, \phi_2, \phi_3, \ldots\}$ represents the unique set of dishes ordered by these tables.

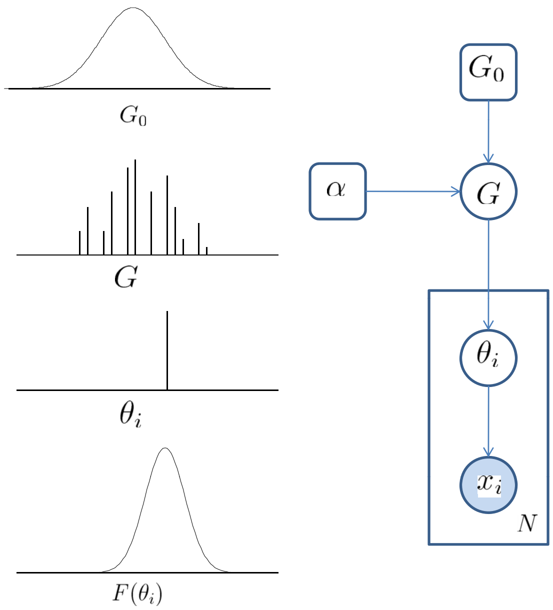


FIGURE 1.2: Graphical model of the DP mixture.

In the Chinese restaurant process, the customers represent the data and the dishes

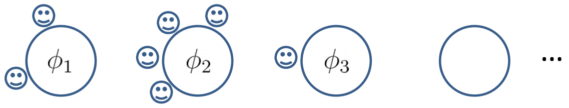


FIGURE 1.3: A typical example of the Chinese restaurant process.

$\{\phi_1, \phi_2, \phi_3, \ldots\}$ are the unique set of parameters underlying the distribution. Assume there are $N-1$ customers having sit in the restaurant and n_k denotes the number of customers sharing the kth table. The data-generating process is as follows:

- The first customer sits at the first table;
- The Nth customer chooses the kth table with a probability proportional to n_k;
- The Nth customer chooses a new table with a probability proportional to α;
- Once a new table is taken, a dish will be drawn from G_0 for it.

Given the data $\{x_1, \ldots, x_N\}$, the joint posterior distribution is for the total number of tables taken by the N customers, the assignment of the customers to each table and also the dishes ordered by the tables. By integrating out the G of Eq.(1.2), the conditional distribution of θ_N given $\theta_1, \ldots, \theta_{N-1}$ is

$$p(\theta_N|\theta_1, \ldots, \theta_{N-1}) = \frac{\alpha}{\alpha + N - 1} G_0 + \sum_{k=1}^{K-1} \frac{n_k}{\alpha + N - 1} \delta_{\phi_k}, \tag{1.3}$$

in which $K - 1$ tables are assumed to be taken by the first $N - 1$ customers and the number of tables is potentially infinite with new customers coming in. Equation (1.3) shows that the joint distribution $p(\theta_1, \ldots, \theta_N)$ is invariant to the permutation of $(x_1, \ldots, x_N)$ and the data are infinitely exchangeable.

Another representation of DP is the stick-breaking process [Set94]. A unit "stick" is broken into infinite proportions and each proportion represents a cluster (see Figure 1.4). A sample drawn from $DP(\alpha G_0)$ can be represented as

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \quad \sum_{k=1}^{\infty} \pi_k = 1, \quad \phi_k \overset{iid}{\sim} G_0$$

$$\pi_k = \prod_{k'=1}^{k-1} (1 - \beta_{k'}) \beta_k, \quad \beta_k \overset{iid}{\sim} Beta(1, \alpha), \tag{1.4}$$

where δ_{ϕ_k} represents a probability measure concentrated at ϕ_k. As mentioned previously, the Dirichlet process is proposed as a measure for distributions that are independent with the permutation of the observations. This is also true for the stick-breaking representation. In (1.4), both $\{\phi_k\}_{k=1}^{K}$ and $\{\beta_k\}_{k=1}^{K-1}$ are i.i.d. drawn, so that the mixture weight $\{\pi_k\}_{k=1}^{K}$ and the atom locations are not functions of any defined covariates. Although such an exchangeable property leads to a simple posterior computation for DP, it is not always true for certain applications associated with data dependence. Therefore, there is growing interest to consider a more general form for nonparametric Bayesian models in recent years.

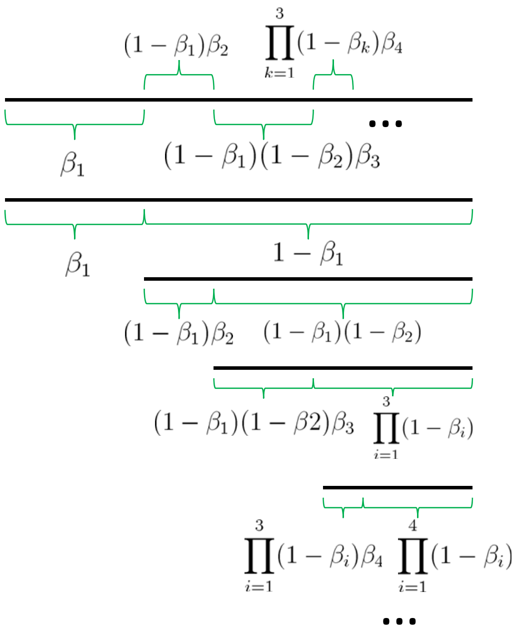

$(1-\beta_1)\beta_2$
$\prod_{k=1}^{3}(1-\beta_k)\beta_4$
$\dots$
β_1
$(1-\beta_1)(1-\beta_2)\beta_3$
β_1
$1-\beta_1$
$(1-\beta_1)\beta_2$
$(1-\beta_1)(1-\beta_2)$
$(1-\beta_1)(1-\beta 2)\beta_3$
$\prod_{i=1}^{3}(1-\beta_i)$
$\prod_{i=1}^{3}(1-\beta_i)\beta_4$
$\prod_{i=1}^{4}(1-\beta_i)$
$\dots$

FIGURE 1.4: An example of stick-breaking process.

1.2 Dependent Nonparametric Probabilistic Models

In this section, we focus on a category of models motivated from two aspects: one starts from the nonparametric way so that the model structure can be adaptively changed with new observations; the other consideration is to define a measure for the distributions over a domain indexed by time, space or some other covariates appended with the data. This motivation makes one naturally think of extending the Dirichlet process to accommodate various dependent distributions. For example, temporal dependence exists in the stock sequential analysis, video target tracking and evolutionary clustering of weblogs, etc; spatial dependence also plays an important role in learning a social network, image segmentation and also many other problems. To exploit the dependence existing in the data structure, it yields a general nonparametric probabilistic model (without the exchangeable assumption) to fit the data with more accurate estimation and prediction; meanwhile a dependent nonparametric clustering model generates a smooth and coherent evolution so that the model can be easily interpretable by researchers.

An important study in this field is the dependent Dirichlet process (DDP) [Mac99, Mac00]. It includes dependence in a collection of distributions with dynamically varying atom locations or stick weights as the functions of data covariates [GS06]. The distribution at a data point $x \in D$ has a general form as follows:

$$G(x) = \sum_{k=1}^{\infty} \pi_k(x) \delta_{\phi_k(x)}, \tag{1.5}$$

in which varying the weights $\pi_k(x)$ captures appearances, disappearances, rise and fall in popularity of the atoms in $G(x)$, while varying the locations $\phi_k(x)$ captures the "atom drift" indicating the atoms change their values in a continuous fashion [AX08].

As we fix the $\pi_k(x) = \pi_k$ for each k, the dependence is introduced only on the atoms, and the model is reduced to a "single-p DDP model" with a stick-breaking representation for the weights π_k. This model has been employed to analyze the variance (ANOVA)-type structure in [IMRM04], and extended to spatial modeling via drawing the atom locations from a Gaussian process [GKM05], leading to a non-stationary and non-normal random process. To make a connection with a more general DDP form, the weights $\pi_k(x)$ are also considered as a function of the covariate x. Underlying this assumption, Duan et al. [DGG07] proposed a general spatial Dirichlet process (GSDP), allowing different sites to choose different random surfaces; Griffin and Steel [GS06] implemented the dependence by inducing an ordering on the random stick variables such that similar covariate values are associated with similar orderings, and thus close distributions; Blei and Lafferty [BL06] extended the latent Dirichlet allocatoin (LDA) [BNJL03] model by introducing the dynamics on both the topics and mixing weights via conditional normal distributions, yielding a evolving topic model for time series.

Another category of dependent nonparametric mixture models are developed in other representation forms of DP. For example, a time-varying Dirichlet process mixture is proposed on an intuitive generalized Polya urn scheme [CDD07], introducing a temporal dependence on both the cluster locations and their weights with a birth/death procedure. The model is easy to understand intuitively but leads to a considerable computation and slow convergence. Another dynamic model representation is based on the recurrent Chinese restaurant process (RCRP) [AX08]. It assumes the data inside each temporal epoch to be fully exchangeable, whereas the temporal order is manifested across epochs; both the popular dishes and the seating plan of the previous epoch will influence the clustering process of the next epoch, explained by the rich-gets-richer phenomenon. Such a partial-exchangeability assumption might reduce the computation cost but the slow convergence issue still exists for a large scale problem.

An alternative way to build a dependent Dirichlet process is to fix the atom locations in the stick-breaking process and generate the mixture weights from a random process dependent on the covariate values. Although one may argue that fixing the atom locations may reduce model flexibility for "atom-drift" phenomena, we don't expect to see poor performance with the model as long as the number of atoms is assumed to be infinite or large. Instead of drawing the random variable β_k associated with each stick k from a beta distribution (see Eq. (1.4)), more flexible forms for generating the random variables are proposed to integrate the covariate-dependence within the model. A typical example is the kernel stick-breaking process (KSBP) [DP07], in which the dependent probability measures are constructed by mixing the predictor locations. Each stick weight is represented in a product form of a beta variable and a bounded kernel function so that the sum of the mixture weights is still equal to one. Another construction is to yield probit transformations of infinite number of normal random variables, replacing the set of beta variables in (1.4) [RD09]. The dependence associated with the covariates is considered as the input of the normal, then transformed to a set of varying mixture weights on the fixed set of atoms. It shows that the process is nonstationary, as one of the limitations induced by using constant atoms. However, this succinct strategy results in simple computation, and is extremely flexible to create different sorts of nonparametric models, including the nonparametric random effects and regression models, etc.

Instead of constructing a covariate-dependent nonparametric random process, it is also interesting to consider a parent-children dependence and build a hierarchical clustering in grouped or nested settings. The hierarchical Dirichlet process (HDP), proposed by Teh et al. [TJBB06], assumes that multiple groups of Dirichlet processes are linked by sharing the same set of atoms and also their popularity. Hence the base measure of each child Dirichlet process is drawn from a common parent Dirichlet process. Under this construction, two representation forms (the Chinese restaurant franchise and a hierarchial stick-breaking process) are discussed and lead to different inference algorithms. Motivated by a more complicated net structure of document topics, the nested Chinese restaurant process [BGJT04] defines a probability distribution on infinitely-deep and infinitely-branching trees. With a Chinese restaurant at the top level, customers on a table can be further decomposed into smaller clusters, generating a hierarchial tree structure. Another nested Dirichlet process [RDG08] is developed in a stick-breaking representation, motivated by a multicenter problem. With each atom defined as a stick-breaking process, two groups may share exactly the same density distribution, including both the atom locations and mixture weights. Although this category of nonparametric model with parent-children dependence cannot yield a set of distributions continuously changed as a function of covariates, it leads to a flexible framework for borrowing information across different observations, and may be employed on a multi-task learning problem.

1.3 Multi-task Learning

Multi-task learning aims to improve the generalization performance of the model estimation and prediction, by sharing data from multiple related tasks. For example, the data might be collected from the same source, or partially share certain common features for different tasks. An easy way to deal with multiple data collections is to pool them together and learn exactly the same model; however, pooling ignores the distinctions and correlation diversity across different data sets. An alternative is to learn each task separately and combine the outputs together. This single-task learning mode leads to an inefficient computation if the number of tasks is large; moreover, the data source is not fully exploited for the model learning. Therefore, how to share the data appropriately according to their relatedness is of interest.

Tasks can be related in different ways [AE06]: assuming that the functions learned should be shared across tasks [Car97,BH03,MP04,YTS05,XLC07] or a lower dimensional embedding is learned in common [BDS03, AE06, RI09]. There are many different ways to realize the transfer learning: learning multiple related functions via a common hyper-prior of Gaussian process [YTS05, CG05, BAW07], developing multi-task kernels based on the feature graph [MP04,LLC07,She08], sharing a hidden layer within neural network models [Car97, BH03], and using the nearest neighbor with a globally weighted distance measure to transfer knowledge selectively [TO96].

Multi-task learning brings two benefits when employed appropriately. First, the model generalization may be improved by borrowing information between tasks, especially as an individual task has limited samples or missing values. For example, Xue et al. [XLC07] performed the multi-task learning in a classification problem and showed a better prediction rate for the testing data; some scholars [AZ05,GLSGFV06, LLC07] exploited the unlabeled data for each task and transferred knowledge across multiple data sets. Furthermore, it is also interesting to obtain certain knowledge, like the sharing structure, to help infer the data source distribution or the correlation between each two tasks.

Motivated by the potential multimodal property associated with multiple data sets, one may naturally consider using the nonparametric techniques to infer model structure. For example, constructing multi-task learning in the Dirichlet process [XLC07], each atom is denoted as a classifier and the sharing mechanism across different tasks is learned in a data-driven manner. Alternatively, a hierarchical model can be built in a group setting with hierarchical Dirichlet process (HDP) [TJBB06], in which each task is learned via a DP mixture model and multiple tasks are linked via a parent Dirichlet process. Under this construction, the model parameters are globally shared, while each task may favor several of those according to the respective weights specifically. The nested Dirichlet process (NDP) [BGJT04,RDG08] can also be extended to a multi-task learning framework, with both the atoms and respective weights being taken by related tasks. Dependent on the specific applications, different algorithms may be chosen accordingly.

1.4 Thesis Organization

This thesis focuses on learning temporal or spatial dependence with Bayesian nonparametric mixture models. Several clustering problems are considered: evolution clustering for music analysis, acoustic data diarization and image segmentation. For each illustrative examples are provided.

The remaining Chapters are organized as follows:

Chapter 2 provides background on time series analysis, with several state-of-art models presented. We start from the hidden Markov model (HMM) [Rab89], focusing on the elementary model structure and its influence on the later work about sequential data analysis. Next, two typical nonparametric hidden Markov models are introduced: one is the infinite hidden Markov model (iHMM) based on the HDP framework [BGR02, TJBB06], with potentially infinite states shared by one HMM; the other one is the hidden Markov model mixture with a DP prior [QPC07], that assuming infinite number of HMMs underlying the observations. We also discuss the dynamic topic model [BL06], with the hidden topics evolved over time. Similar to the model described in Chapter 3, the dynamic topic model explicitly builds temporal dependence on the observations, instead of the hidden states like HMM.

Chapter 3 develops a nonparametric evolving-clustering model, the dynamic hierarchical Dirichlet process (dHDP). The data collected at any time point are represented via a mixture associated with an appropriate underlying model, in the framework of HDP. The statistical properties of data collected at consecutive time points are linked via a random parameter that controls their probabilistic similarity. The sharing mechanisms of the time-evolving data are derived, and a relatively simple Markov Chain Monte Carlo sampler is developed. As the model employed for music analysis, each music piece is represented in terms of a sequence of discrete observations, and the sequence is modeled using a hidden Markov model (HMM) with time-evolving parameters. The dHDP imposes the belief that observations that are temporally proximate are more likely to be drawn from HMMs with similar parameters, while also allowing for "innovation" associated with abrupt changes in the music texture. Detailed examples are presented on several pieces, with comparisons to other models and a conventional music-theoretic analysis.

Chapter 4 discusses modeling spatial dependence in a nonparametric manner. Motivated by a time-evolving model, spatially dependent information may also be included within the model prior, like the Markov random field [GG84, BVZ98]. Instead of setting the neighborhood given a graph, it is more attractive to consider learning the partition with a data-driven method. Two nonparametric models, the generalized spatial Dirichlet process and kernel stick-breaking process, are discussed separatively, as a motivation for a new model proposed in the following part.

Chapter 5 presents an innovative nonparametric clustering model, the logistic stick-breaking process (LSBP), for general spatially- or temporally- dependent data. The sticks in the LSBP are realized via multiple logistic regression functions, with shrinkage priors employed to favor contiguous and spatially localized segments. Efficient variational Bayesian inference is derived, and comparisons are made to related techniques. Experimental analysis is performed for both audio waveforms and images.

Chapter 6 focuses on the multi-task learning for the simultaneous processing of multiple data sets, yielding a hierarchical logistic stick-breaking process (H-LSBP). The model parameters within the H-LSBP are shared across the multiple learning tasks while different tasks are assumed to be conditionally independent. The new framework is applied on joint image segmentation, with illustrative examples followed in the end.

Chapter 7 concludes the thesis and its contributions. Several possible directions are also provided for future work.

2

Probabilistic Time Series Models

This chapter provides background on time series analysis, with several state-of-art models being presented. Section 2.1 reviews the hidden Markov model (HMM) [Rab89] with its model structure and applications; Section 2.2 presents two nonparametric HMMs: Infinite HMM [BGR02,TJBB06] and HMM mixture with a DP prior [QPC07]. The first model assumes infinitely potential states within one HMM; the second one is constituted of infinite HMMs to fit the data. Both of the models are employed on music analysis in Chapter 3. Section 2.3 discusses the dynamic topic model [BL06] and the study of time-evolving topics.

2.1 Hidden Markov Model

The standard tool for analysis of sequential data is the hidden Markov model (HMM) [BP66, BPSW]. It is a doubly embedded stochastic process with an underlying hidden stochastic process of state transition and an observed one producing the sequential observations [Rab89]. The observations can either be continuous or represented as discrete symbols from a codebook. For the discrete sequence of interest, given an observation sequence $\mathbf{x} = \{x_t\}_{t=1}^{T}$ with $x_t \in \{1, \ldots, M\}$ (M is the discrete alphabet size), the corresponding hidden state sequence is $\mathbf{S} = \{s_t\}_{t=1}^{T}$, from which $s_t \in \{1, \ldots, I\}$ (I is the total number of states).

We consider an underlying (hidden) first order of Markov chain associated with the states, so that the current state s_t depends only on the one at the previous time, i.e., $p(s_t|s_{t-1}, \ldots, s_1) = p(s_t|s_{t-1})$. The number of unique states I is unknown and may be fixed at the beginning. Under the assumption, an HMM is represented by parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, defined as

- $\mathbf{A} = \{a_{\rho\xi}\}$, $a_{\rho\xi} = Pr(s_{t+1} = \xi | s_t = \rho)$: state transition probability;
- $\mathbf{B} = \{b_{\rho m}\}$, $b_{\rho m} = Pr(x_t = m | s_t = \rho)$: emission probability;
- $\boldsymbol{\pi} = \{\pi_\rho\}$, $\pi_\rho = Pr(s_1 = \rho)$: initial state distribution.

Given the model parameters, the data generation can be summarized in Figure 2.1:

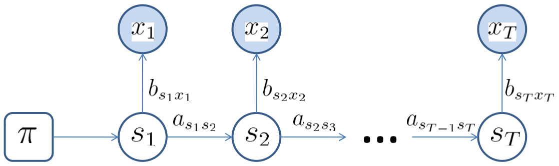


FIGURE 2.1: Graphical model of HMM.

1. at the beginning with $t = 1$, s_1 is generated from the initial state distribution Mult($\boldsymbol{\pi}$);

2. according to the state s_1, x_1 is chosen from the distribution of observation emission, Mult($b_{s_1\cdot}$) ($b_{s_1\cdot}$ represents the row of $\mathbf{B}$ matrix with the index of s_1);

3. from the state transition, s_t is generated from the corresponding distribution Mult($a_{s_t\cdot}$) ($a_{x_t\cdot}$ represents the row of $\mathbf{A}$ matrix with the index of s_t);

4. x_t is chosen from the distribution Mult($b_{s_t\cdot}$);

5. repeat step 3 $\sim$ 4 for time $t + 1$ until the end

Based on the model parameters, the data likelihood is obtained by summing the joint probability over all possible state sequences:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{S}} \pi_{s_1} \prod_{t=1}^{T-1} a_{s_t,s_{t+1}} \prod_{t=1}^{T} b_{s_t,x_t} \tag{2.1}$$

However, it leads to the an unfeasible computation, being ergodic for the whole state space at every time.

A more efficient computation method is the forward-backward procedure [BE67]. For each time, the sufficient statistics $\alpha_t(\rho) = P(x_1, \ldots, x_t, s_t = \rho|\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ and $\beta_t(\rho) = P(x_{t+1}, \ldots, x_T|s_t = \rho, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ with $\rho = 1, \ldots, I$ are calculated.

In the forward way:

- from $t = 1$, $\alpha_1(\rho)$ is initialized as $\pi_\rho b_{\rho x_1}$;
- for the following time, $\alpha_{t+1}(\xi) = \left[\sum_{\rho=1}^{I} \alpha_t(\rho) a_{\rho\xi}\right] b_{\xi x_{t+1}}$ in an inductive calculation;
- until the end $P(\mathbf{x}|\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) = \sum_{\rho=1}^{I} \alpha_T(\rho)$.

In the backward way:

- from the end T, $\beta_T(\rho) = 1$ for $\rho = 1, \ldots, I$;
- for each $t - 1$, $\beta_{t-1}(\xi) = \sum_{\rho=1}^{I} a_{\xi\rho} b_{\rho x_t} \beta_t(\rho)$;
- similar to the forward calculation, $P(\mathbf{x}|\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) = \sum_{\xi=1}^{I} \beta_1(\xi)$.

Based on both $\{\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t\}_{t=1}^{T}$, we can calculate the the best state sequence and optimize the HMM parameters with a reestimation procedure [Rab89].

Hidden Markov models have been widely used for time-series analysis, such as speech recognition, gesture recognition, music classification and motion detection. However, the model definition suffers from an important limitation: the model structure has to be specified in advance, making it difficult to avoid over or underfitting [BGR02]. Under a new framework of nonparametric techniques, scholars proposed a more flexible model structure: allowing to automatically control the model complexity of HMM according to the data.

2.2 Nonparametric Hidden Markov Model

There are two ways to define an HMM with "infinite" parameters. The first one is assuming that there are infinite number of hidden states within the HMM and the model parameters $\{\boldsymbol{\pi},\mathbf{A},\mathbf{B}\}$ is not specified with a fixed dimension; the second one is to fit the data with an HMM mixture associated with infinite components while each HMM has a fixed model structure.

2.2.1 Infinite HMM

An infinite hidden Markov model (iHMM) [BGR02, TJBB06] denotes an HMM with an infinite number of hidden states. Each row of the state transition matrix is modeled with a DP prior and the multiple rows are linked with another DP. The whole model structure proposed fits the hierarchical Dirichlet process (HDP) [TJBB06].

HDP was first proposed for model-based clustering of grouped data. The hierarchical specification is summarized as follows:

$$\begin{aligned} G_0|\gamma, H &\sim DP(\gamma H), \\ G_j|\alpha, G_0 &\sim DP(\alpha G_0). \end{aligned} \tag{2.2}$$

Two key properties that HDP displays are attractive: the global DP introduces a discrete base G_0 shared by each child-DP, yielding the desired sharing of atoms across groups; the nonparametric priors adopted allows infinite number of components constituted to serve for each group G_j. An alternative representing form of HDP is based on stick-breaking construction:

$$\begin{aligned} \boldsymbol{\beta}|\gamma &\sim GEM(\gamma), \\ \boldsymbol{\pi}_j|\alpha, \boldsymbol{\beta} &\sim DP(\alpha\boldsymbol{\beta}), \\ \phi_k|H \sim H \quad &\text{for} \quad k = 1, \ldots, \infty. \end{aligned} \tag{2.3}$$

Combining the representation within (2.2), $G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$ and $G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}$.

Considering the HMM again, if we regard the rows of the state transition matrix as multiple groups and each state as the index of components, an infinite HMM can be constructed based on HDP as in Figure 2.2. The observed variables $x_1, \ldots, x_T$ represent one sequence and the embedding hidden states are $s_1, \ldots, s_T$. Here an initial state s_0 is assumed to be known in advance. There are countably infinite unique state values, with index $k = 1, \ldots, \infty$, that can be taken by s_t for each time; corresponding to each state index k, there is a state transition distribution π_k drawn from $DP(\alpha\boldsymbol{\beta})$. Given the previous state s_{t-1}, s_t is generated from $\text{Mult}(\boldsymbol{\pi}_{s_{t-1}})$, then x_t will be generated from the distribution $F(\phi_{s_t})$. The observation parametric form $F(\cdot)$ can either be a multinomial or Gaussian distribution, depending on whether x_t is discrete or continuous.

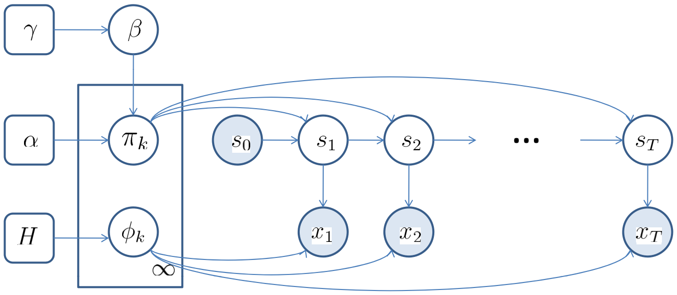


FIGURE 2.2: Graphical model for iHMM.

Several MCMC sampling schemes have been introduced for the iHMM in [TJBB06], including the Chinese restaurant franchise representation, the augmented, and direct assignment representations; alternatively we can use a variational Bayesian inference based on the stick-breaking representation [NCD07]. For all of the inference algorithms, the parameters $\{\boldsymbol{\pi}_k\}_{k=1}^{\infty}$ in Figure 2.2 record the information of state transition for each time and $\{\phi_k\}_{k=1}^{\infty}$ concludes the emission probability. Hence, they represent the state transition matrix $\mathbf{A}$ and observation emission distribution $\mathbf{B}$. The number of times state k is chosen first is also calculated for all of the sequences to update the initial state distribution $\boldsymbol{\pi}$ accordingly.

2.2.2 HMM Mixture with a DP Prior

Another representative nonparametric hidden Markov model is the HMM mixture with a DP prior [QPC07], motivated by clustering multiple sequences with various temporal statistics. Similar to the infinite Gaussian mixture model (iGMM) [Ras00], N random variables $\{\theta_n\}_{n=1}^{N}$ are drawn from G, and G itself is a random measure drawn from $DP(\alpha G_0)$. The clustering property of DP encourages sharing parameters, and this naturally reveals the proper number of mixture components [QPC07]. For an HMM mixture, the observation $\mathbf{x}_n$ represents the nth sequence and there are N sequences in total; each θ_n represents the finite-state HMM generating the sequence $\mathbf{x}_n$. With these assumptions, the model is summarized as follows:

$$\begin{aligned}\mathbf{x}_n|c_n,\{\theta_k^*\}_{k=1}^{\infty} &\sim F(\theta_{c_n}^*)\\ c_n|\mathbf{p} &\overset{i.i.d}{\sim} Mult(\mathbf{p})\\ \mathbf{p}|\alpha &\sim GEM(\alpha)\\ \theta_k^*|G_0 &\sim G_0\end{aligned} \tag{2.4}$$

Here c_n is an indicator variable to denote which HMM from $\{\theta_k^*\}_{k=1}^{\infty}$ the data $\mathbf{x}_n$ is drawn from.

For this model, even though each component has a parametric form with a fixed number of states, the model still has infinite number of states in total, as a result of infinite HMMs constituted. Both the infinite HMM model and the HMM mixture with a DP prior have been applied on music analysis [NCD07, QPC07], with each music piece trained by one iHMM or an HMM mixture; the similarity between two music pieces can be measured by a kernel distance [AP02, QPC07] between the respective models for different pieces. Compared with the iHMM, one distinct utility of the HMM mixture is that it allows one to analyze different segments of a given piece by studying the membership of each subsequence associated with different atoms;

however, the exchangeable assumption underlying the indicator variables c_n does not match with the coherence property of music and makes it difficult to explain the music's structure with such a model.

Another limitation of the HMM is that the model does not explicitly assume a time-evolving permutation of parameters underlying the observations. The temporal dependence is built on a probabilistic state transition via a Markov chain, which is still not strong enough to reach a smooth evolutionary-clustering result. This motivates scholars to extend classical state space model to a dynamic topic model with parameters dynamically evolving along the time.

2.3 Dynamic Topic Model

The dynamic topic model [BL06] is proposed to capture the evolution of topics in a sequentially organized corpus of documents. Each document is composed of multiple topics and each topic is defined as a multinomial distribution on a set of words. In the traditional topic model with Latent Dirichelt Allocation (LDA) [BNJL03], the data generation starts from picking a topic from the topic distribution (defined as a multinomial), then draws a word from the selected topic; all the words are assumed exchangeable and the topics inferred from each document will not be influenced by the order of the words. For the dynamic topic model, a set of document collection is provided and the documents are reordered according to their appended temporal information. For example, a set of academic articles from *Science* are organized as the order of their publishing year. Hence it is interesting to explicitly model the dynamics of the underlying topics.

There are two factors that a topic model may consider for dynamic evolution: the topics themselves and topic proportion distributions. In the dynamic topic model [BL06], both of them are considered to evolve with Gaussian noise from the previous time-stamp $t-1$ to the current time t, as shown in Figure 2.3. The parameter at the previous time $t-1$ is the expectation for the distribution of the parameter at the next time t, and the correlation of the samples at adjacent times is controlled through adjusting the variance of the conditional distribution. Instead of drawing the topic proportions θ_t from a Dirichlet distribution like LDA [BNJL03], the model draws it from a logistic normal with mean α_t to express uncertainty over proportions. A similar technique is also applied on the word distributions associated with each topic. The generative process for the documents collected at t is summarized as follows:

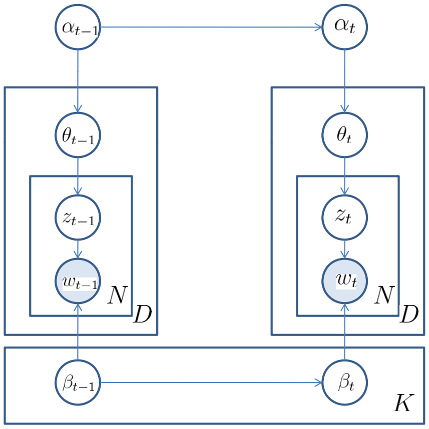


FIGURE 2.3: Graphical model for Dynamic topic model.

- Assume D documents at each time slice and N words in each document;
- Define K topics β for each time and drawn $\beta_t|\beta_{t-1} \sim N(\beta_{t-1}, \sigma^2 I)$;
- Draw $\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \delta^2 I)$;
- For each document:

  1. Draw $\eta \sim N(\alpha_t, a^2 I)$ and $\theta_k = \frac{exp(\eta_k)}{\sum_{k'} exp(\eta_{k'})}$;
  2. For each word:

     a. Draw the topic indicator $z \sim Mult(\theta)$;

b. Draw word $w \sim Mult(\pi(\beta_{t,z}))$, where $\pi(\beta_{t,k})_w = \frac{exp(\beta_{t,k,w})}{\sum_{w'} exp(\beta_{t,k,w'})}$.

By chaining both the topics and topic proportion distributions, the topics smoothly evolve over time [BL06] and the model is proved to yield a more accurate prediction. Unfortunately, the non-conjugate form of the conditional distribution requires approximations in the model inference. Considering the above discussions, we expect to build a dynamic model with the following characteristics: (i) the number of model parameters is not known *a priori* and can be inferred from the data automatically; (ii) the temporal information should be considered to generate a smooth time-evolving clustering; (iii) a conjugate model form is desired for an easier posterior inference.

3

The Dynamic Hierarchical Dirichlet Process

The Dirichlet process (DP) mixture model [EW95] has been widely used to perform density estimation and clustering, by generalizing finite mixture models to (in principle) infinite mixtures. In order to "share statistical strength" across different groups of data, the hierarchical Dirichlet process (HDP) [TJBB06] has been proposed to model the dependence among groups through sharing the same set of discrete parameters ("atoms"), and the mixture weights associated with different atoms are varied as a function of the data group. In the HDP, it is assumed that the data groups are exchangeable. However, in many real applications, such as seasonal market analysis and gene investigation for disease, data are measured in a sequential manner, and there is information in this temporal character that should ideally be exploited; this violates the aforementioned assumption of exchangeability.

Recently Dunson [Dun06] proposed a Bayesian dynamic model to learn the latent trait distribution through a mixture of DPs, in which the latent variable density changes dynamically in location and shape across levels of predictors. This dynamic structure is considered here to extend HDP to incorporate time dependence, and has the following features: (*i*) two data samples drawn at proximate times have a higher probability of sharing the same underlying model parameters (atoms) than parameters drawn at disparate times; and (*ii*) there is a possibility that temporally distant data samples may also share model parameters, thereby accounting for possible distant repetition in the data.

3.1 Dynamic Hierarchical Dirichlet Process

3.1.1 Nonparametric Bayesian Dynamic Structure

Similar to HDP, we again consider J data sets but now using an explicit assumption that the data sets are collected sequentially, with $\{x_{1,i}\}_{i=1,\ldots,N_1}$ collected first, $\{x_{2,i}\}_{i=1,\ldots,N_2}$ collected second, and with $\{x_{J,i}\}_{i=1,\ldots,N_J}$ collected last. Since our assumption is that a time evolution exists between adjacent data groups, the distribution G_{j-1}, from which $\{\boldsymbol{\theta}_{j-1,i}\}_{i=1,\ldots,N_{j-1}}$ are drawn, is likely related to G_j, from which $\{\boldsymbol{\theta}_{j,i}\}_{i=1,\ldots,N_j}$ are drawn.

To specify explicitly the dependence between G_{j-1} and G_j, Dunson [Dun06] proposed a Bayesian dynamic mixture DP (DMDP), in which G_j shares features with G_{j-1} but some innovation may also occur. The DMDP has the drawback that mixture components can only be added over time, so that one ends up with more components at later times as an artifact of the model.

In the dHDP, we have

$$G_j = (1 - \tilde{w}_{j-1})G_{j-1} + \tilde{w}_{j-1}H_{j-1} \tag{3.1}$$

where $G_1 \sim DP(\alpha_{01}, G_0)$, H_{j-1} is called an innovation distribution drawn from $DP(\alpha_{0j}, G_0)$, and $\tilde{w}_{j-1} \sim Be(a_{w(j-1)}, b_{w(j-1)})$. In this way, G_j is modified from G_{j-1} by introducing a new innovation distribution H_{j-1}, and the random variable $\tilde{w}_{j-1}$ controls the probability of innovation (*i.e.*, it defines the mixture weights). As a result, the relevant atoms adjust with time, and it is probable that proximate data will share the same atoms, but with the potential for transient innovation.

Additionally, we assume that $G_0 \sim DP(\gamma, H)$ as in the HDP to enforce that G_0 is discrete, which manifests another important aspect of the dynamic HDP: the same atoms are used for *all* G_j, but with different time-evolving weights. Consequently, the model encourages sharing between temporally proximate data, but it is also possible to share between data sets widely separated in time.

Providing now more model details, the discrete base distribution drawn from $DP(\gamma, H)$ may be expressed as:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\boldsymbol{\theta}_k^*} \tag{3.2}$$

where $\{\boldsymbol{\theta}_k^*\}_{k=1,2,\ldots,\infty}$ are the global parameter components (atoms), drawn independently from the base distribution H and $\{\beta_k\}_{k=1,2,\ldots,\infty}$ are drawn from a stick-breaking process $\boldsymbol{\beta} \sim Stick(\gamma)$, defined as:

$$\beta_k = \tilde{\beta}_k \prod_{l<k}(1 - \tilde{\beta}_l) \qquad \tilde{\beta}_k \overset{iid}{\sim} Be(1, \gamma) \tag{3.3}$$

We also have J groups of data. G_j represents the prior for the mixture distribution associated with the global components in group j, H_{j-1} represents the associated prior for the innovation mixture distribution, and this yields the explicit priors used in (3.1):

$$G_1 = \sum_{k=1}^{\infty} \pi_{1,k} \delta_{\boldsymbol{\theta}_k^*}, H_1 = \sum_{k=1}^{\infty} \pi_{2,k} \delta_{\boldsymbol{\theta}_k^*}, \ldots, H_{J-1} = \sum_{k=1}^{\infty} \pi_{J,k} \delta_{\boldsymbol{\theta}_k^*} \tag{3.4}$$

where the different weights $\boldsymbol{\pi}_j$ are independent given $\boldsymbol{\beta}$ since $G_1, H_1, \ldots, H_{J-1}$ are independent given G_0; the relationship between $\boldsymbol{\pi}_j$ and $\boldsymbol{\beta}$ is proven [TJBB06] to be

$$\boldsymbol{\pi}_j | \alpha_{0j}, \boldsymbol{\beta} \sim DP(\alpha_{0j}, \boldsymbol{\beta}) \tag{3.5}$$

To further develop the dynamic relationship from G_1 to G_J, we extend the mixture structure in (3.1) from group to group:

$$\begin{aligned} G_j &= (1-\tilde{w}_{j-1})G_{j-1} + \tilde{w}_{j-1}H_{j-1} \\ &= \prod_{l=1}^{j-1}(1-\tilde{w}_l)G_1 + \sum_{l=1}^{j-1}\{\prod_{m=l+1}^{j-1}(1-\tilde{w}_m)\}\tilde{w}_l H_l \\ &= w_{j1}G_1 + w_{j2}H_1 + \ldots + w_{jj}H_{j-1} \end{aligned} \tag{3.6}$$

where $w_{jl} = \tilde{w}_{l-1}\prod_{m=l}^{j-1}(1-\tilde{w}_m)$, for $l = 1, 2, \ldots, j$, with $\tilde{w}_0 = 1$. It can be easily verified that $\sum_{l=1}^{j} w_{jl} = 1$ for each $\mathbf{w}_j$, which is the prior probability that the data in group j will be drawn from the mixture distribution: $G_1, H_1, \ldots, H_{j-1}$. If all $\tilde{w}_j = 0$, all of the groups share the same mixture distribution G_1 and the model reduces to a Dirichlet mixture model, and if all $\tilde{w}_j = 1$ the model reduces to the HDP. Therefore, the dynamic HDP is more general than both DP and HDP, with each a special case. A visual representation of the model is depicted in Figure 3.1.

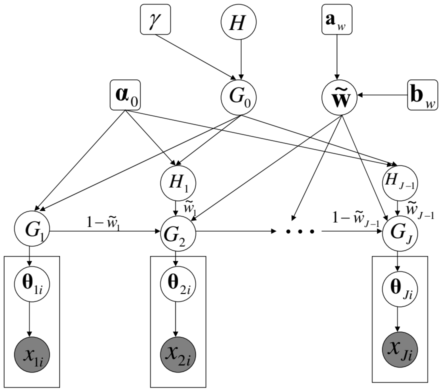


FIGURE 3.1: General graphical model for the dynamic HDP.

According to (3.6), the observation $x_{j,i}$ will choose a mixture distribution from $\boldsymbol{\pi}_{1:j}$ based on Mult$(\mathbf{w}_j)$ to be drawn from the global parameter components $\{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty}$. We let $r_{j,i}$ be a variable to indicate which mixture distribution is taken from $\boldsymbol{\pi}_{1:j}$ to draw the observation $x_{j,i}$; $z_{j,i}$ is a parameter component indicator variable. An alternative form of the dHDP model is represented as:

$$\begin{gathered}\boldsymbol{\theta}_k^*|H \sim H, \quad \boldsymbol{\beta}|\gamma \sim Stick(\gamma) \\ \tilde{w}_j|a_{wj}, b_{wj} \sim Be(\tilde{w}_j|a_{wj}, b_{wj}), \quad r_{j,i}|\tilde{\mathbf{w}} \sim \mathbf{w}_j \\ \boldsymbol{\pi}_j|\alpha_{0j}, \boldsymbol{\beta} \sim DP(\alpha_{0j}, \boldsymbol{\beta}), \quad z_{j,i}|\boldsymbol{\pi}_{1:j}, r_{j,i} \sim \boldsymbol{\pi}_{r_{j,i}} \\ x_{j,i}|z_{j,i}, (\boldsymbol{\theta}_k^*)_{k=1}^{\infty} \sim F(\boldsymbol{\theta}_{z_{j,i}}^*),\end{gathered} \tag{3.7}$$

and a graphical representation is shown in Figure 3.2, in which we add a gamma prior for γ and for the components of the vector $\boldsymbol{\alpha}_0$: $Pr(\gamma) = Ga(\gamma; \gamma_{01}, \gamma_{02})$ and $Pr(\boldsymbol{\alpha}_0) = \prod_{j=1}^J Ga(\alpha_{0j}; c_0, d_0)$. The form of the parametric model $F(\cdot)$ may be varied depending on the application.

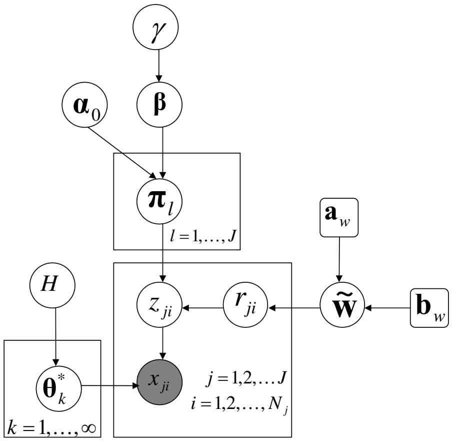


FIGURE 3.2: Graphical representation of the dHDP from a stick-breaking view.

3.1.2 Sharing Properties

To obtain insight into the dependence structure induced by the dHDP proposed in 3.1.1, this section presents some basic properties. Suppose G_0 is a probability measure on $(\Omega, \mathcal{B})$, with Ω the sample space of $\boldsymbol{\theta}_{ji}$ and $\mathcal{B}(\Omega)$ the Borel σ-algebra of subsets of Ω. Then for any $B \in \mathcal{B}(\Omega)$

$$\big(G_j(B)|G_{j-1}, \tilde{w}_j\big) \overset{\mathcal{D}}{=} G_{j-1}(B) + \Delta_j(B), \tag{3.8}$$

where $\Delta_j(B) = \tilde{w}_{j-1}\big\{H_{j-1}(B) - G_{j-1}(B)\big\}$ is the random deviation from G_{j-1} to G_j.

Theorem 1. Under the dHDP (3.6), for any $B \in \mathcal{B}(\Omega)$ we have:

$$E\big\{\Delta_j(B)|G_{j-1}, \tilde{w}_{j-1}, G_0, \alpha_{0j}\big\} = \tilde{w}_{j-1}\big\{G_0(B) - G_{j-1}(B)\big\}, \tag{3.9}$$

$$V\big\{\Delta_j(B)|G_{j-1}, \tilde{w}_{j-1}, G_0, \alpha_{0j}\big\} = \tilde{w}_{j-1}^2 \frac{G_0(B)\big(1 - G_0(B)\big)}{(1 + \alpha_{0j})}. \tag{3.10}$$

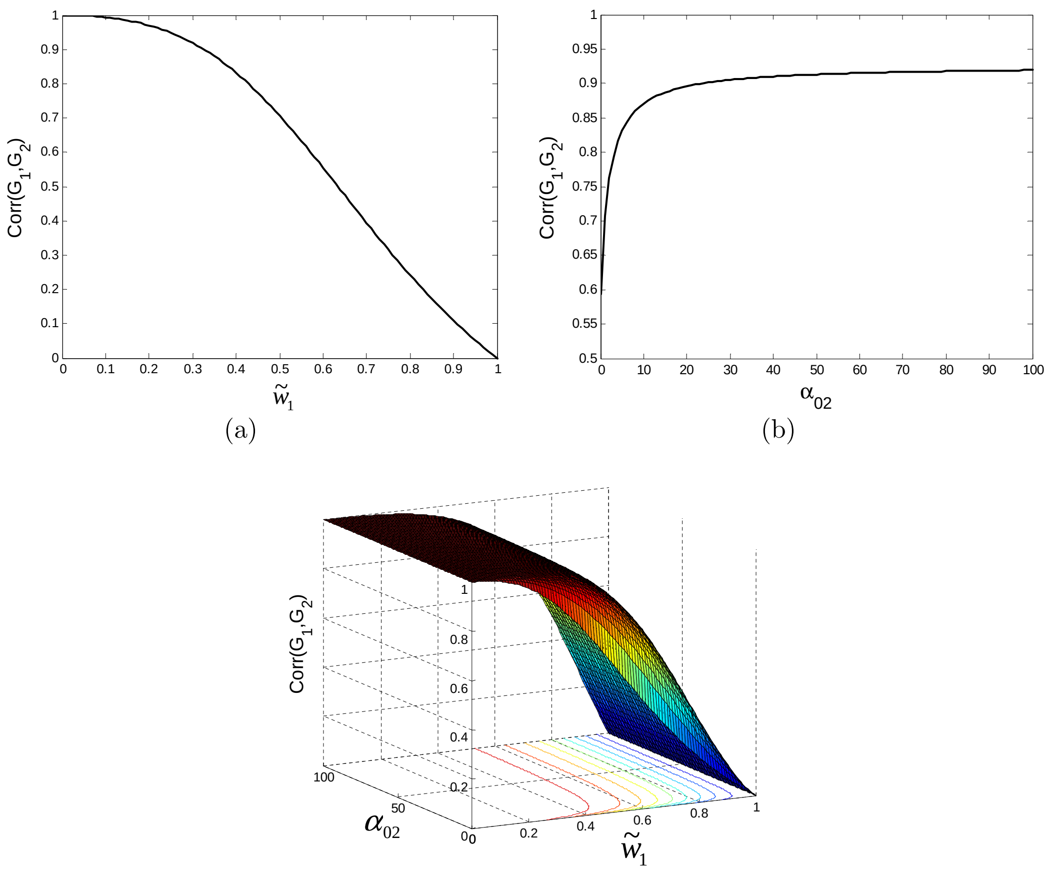
The proof is straightforward and is omitted. According to Theorem 1, given the previous mixture measure G_{j-1} and the global mixture G_0, the expectation of the deviation from G_{j-1} to G_j is controlled by $\tilde{w}_{j-1}$. Meanwhile, the variance of the deviation is related with both $\tilde{w}_{j-1}$ and the precision parameters α_{0j} given G_0. In the limiting case, we obtain the following: If $\tilde{w}_{j-1} \to 0$, $G_j \to G_{j-1}$; If $G_{j-1} \to G_0$, $E\big(G_j(B)|G_{j-1}, \tilde{w}_{j-1}, G_0, \alpha_{0j}\big) \to G_{j-1}(B)$; If $\alpha_{0j} \to \infty$, $V\big(\Delta_j(B)|G_{j-1}, \tilde{w}_{j-1}, G_0, \alpha_{0j}\big) \to 0$.

Theorem 2. Conditional on the mixture weights $\mathbf{w}$, the correlation coefficient of the measures between two adjacent groups $G_{j-1}(B)$ and $G_j(B)$ for $j = 2, \ldots, J$ is

$$\begin{aligned} Corr(G_{j-1}, G_j) =& \frac{E\big\{G_j(B)G_{j-1}(B)\big\} - E\big\{G_j(B)\big\}E\big\{G_{j-1}(B)\big\}}{\Big[V\big\{G_j(B)\big\}V\big\{G_{j-1}(B)\big\}\Big]^{1/2}} \\ =& \frac{\sum_{l=1}^{j-1} \frac{w_{jl}w_{j-1,l}}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)}{\Big[\sum_{l=1}^{j} \frac{w_{jl}^2}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)\Big]^{1/2}\Big[\sum_{l=1}^{j-1} \frac{w_{j-1,l}^2}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)\Big]^{1/2}} \end{aligned} \tag{3.11}$$

The proof is given in the Appendix A. Due to the lack of dependence on B, Theorem 2 provides a useful expression for the correlation between the measures, which can provide insight into the dependence structure.

To study how the correlation depends on $\tilde{\mathbf{w}}$ and $\boldsymbol{\alpha}_0$, we focus on $Corr(G_1, G_2)$ and (i) in Figure 3.3 (a) we plot the correlation coefficient $Corr(G_1, G_2)$ as a function



(a)

(b)

(c)

FIGURE 3.3: (a) $Corr(G_1, G_2)$ as a function of $\tilde{w}_1$ with γ and $\boldsymbol{\alpha}$ fixed. (b) $Corr(G_1, G_2)$ as a function of α_{02}, with γ, α_{01} and $\mathbf{w}$ fixed. (c) $Corr(G_1, G_2)$ as a function of both $\tilde{w}_1$ and α_{02}, with the values of γ and α_{01} fixed.

of $\tilde{w}_1$, with the precision parameters γ and $\boldsymbol{\alpha}_0$ fixed at one; (*ii*) in Figure 3.3 (b) we plot $Corr(G_1, G_2)$ as a function of α_{02}, with $\tilde{w}_1 = 0.5$, $\alpha_{01} = 1$ and $\gamma = 10$; (*iii*) in Figure 3.3 (c) we consider the plot of $Corr(G_1, G_2)$ as a function of both the variables of $\tilde{w}_1$ and α_{02} given fixed values of $\gamma = 10$ and $\alpha_{01} = 1$. It is observed that the correlation between adjacent groups increases with smaller $\tilde{w}$ and larger $\boldsymbol{\alpha}_0$. If we assume that $\alpha_{0l} = \alpha$ for $l = 1, \ldots, j$, then the correlation coefficient has the simple form

$$Corr(G_{j-1}, G_j) = \frac{\sum_{l=1}^{j-1} w_{jl} w_{j-1,l}}{\left\{ \sum_{l=1}^{j} w_{jl}^2 \right\}^{1/2} \left\{ \sum_{l=1}^{j-1} w_{j-1,l}^2 \right\}^{1/2}}. \tag{3.12}$$

3.1.3 Comparisons with Alternative Models

It is useful to consider relationships between the proposed dHDP and other dynamic nonparametric Bayes models. A particularly relevant connection is to dependent Dirichlet processes (DDPs) [Mac99], which provide a class of priors for dependent collections of random probability measures indexed by time, space, or predictors. DDPs were applied to time series settings by Rodriguez and Ter Horst [RH08]. Dynamic DDPs have the property that the probability measure at a given time is marginally assigned a Dirichlet process prior, while allowing for dependence between the measures at different times through a stochastic process in the weights and/or atoms. Most of the applications have relied on the assumption of fixed weights, while allowing the atoms to vary according to a stochastic process. Varying weights is well motivated in some applications, such as the music analysis due to repetition in the music piece, and can be accommodated by the order-based DDP [GS06] and the local Dirichlet process [CD09b]. However, these approaches do not naturally allow long-range dependence and can be complicated to implement. Simpler approaches were proposed by Caron et al. [CDD$^+$08] using dynamic linear models with Dirichlet process components and by Caron, Davy, and Doucet [CDD07] using a dynamic modification of the DP Polya urn scheme. Again, these approaches do not automatically allow long range dependence.

The dHDP can alternatively be characterized as a process that first draws a latent collection of distributions, $\mathcal{H} = \{G_1, H_1, \ldots, H_{J-1}\}$, from an HDP, with the HDP providing a special case of the DDP framework. The parameter vectors of the jth group, is then associated with the lth distribution in the collection $\mathcal{H}$ with probability w_{jl}. This specification simplifies posterior computation and interpretation, while allowing a flexible long range dependence structure. An alternative to the HDP would be to choose a nested Dirichlet process (nDP) [RDG08] prior for the collection $\mathcal{H}$. The nDP would allow clustering of the component distributions within $\mathcal{H}$; distributions within a cluster are identical while distributions in different clusters have different atoms and weights. This structure also accommodates long range dependence but in a very different manner that may be both more difficult to interpret and more flexible in allowing different atoms at different times.

3.1.4 Posterior Computation

There are two commonly used Gibbs sampling strategies for posterior computation in DPMs. The first relies on marginalizing out the random measure through use of the Polya urn scheme [BM96], while the second relies on truncations of the stick-breaking representation [IJ01]. As it is not straightforward to obtain a generalized urn scheme for the dHDP, we rely on the latter approach, which is commonly referred to as the blocked Gibbs sampler. The primary conditional posterior distributions used in implementing this approach are listed as follows:

1. The update of $\tilde{w}_l$, for $l = 1, \ldots, J-1$ from its full conditional posterior distribution, has the simple form

$$(\tilde{w}_l|\cdots) \sim Be(a_w + \sum_{j=l+1}^{J} n_{j,l+1}, b_w + \sum_{j=l+1}^{J}\sum_{h=1}^{l} n_{jh}) \tag{3.13}$$

where $n_{jh} = \sum_{i=1}^{N_j} \delta(\mathbf{r}_{ji,h} = 1)$, $\{\mathbf{r}_{ji}\}_{j=1,\ldots,J;i=1,\ldots,N_j}$ are indicator vectors and $\delta(r_{ji,h} = 1)$ denotes that $\boldsymbol{\theta}_{ji}$ is drawn from the hth component distribution in (3.6). In (3.13) and in the results that follow, for simplicity, the distributions $Be(a_{wj}, b_{wj})$ are set with fixed parameters $a_{wj} = a_w$ and $b_{wj} = b_w$ for all time samples. The function $\delta(\cdot)$ equals 1 if $(\cdot)$ is true and 0 otherwise.

2. Assume the truncation level is K for each $\{\boldsymbol{\pi}_l\}_{l=1}^{J}$ in Eq. (3.5), and $\pi_{lk} = \tilde{\pi}_{lk}\prod_{m=1}^{k-1}(1-\tilde{\pi}_{lm})$. The full conditional distribution of $\tilde{\pi}_{lk}$, for $l = 1, \ldots, J$ and $k = 1, \ldots, K$, is updated under the conjugate prior $\tilde{\pi}_{lk} \sim Be\Big[\alpha_{0l}\beta_k, \alpha_{0l}(1 - \sum_{m=1}^{k}\beta_m)\Big]$,

which is specified in [TJBB06]. The likelihood function associated with each $\tilde{\boldsymbol{\pi}}_l$ is proportional to $\prod_{k=1}^{K} \pi_{lk}^{\sum_{j=l}^{J} \sum_{i=1}^{N_j} \delta(r_{ji,l}=1, z_{ji,k}=1)}$, where $\mathbf{z}_{ji}$ is another indicator vector, with $z_{ji,k} = 1$ if the observation $\mathbf{x}_{ji}$ is allocated to the k^{th} atom ($\boldsymbol{\theta}_{ji} = \boldsymbol{\theta}_k^*$) and $z_{ji,k} = 0$ otherwise. Then the conditional posterior of $\tilde{\pi}_{lk}$ has the form

$$(\tilde{\pi}_{lk}|\cdots) \sim Be\Bigg[\alpha_{0l}\beta_k + \sum_{j=l}^{J}\sum_{i=1}^{N_j} \delta(r_{ji,l}=1, z_{ji,k}=1),$$

$$\alpha_{0l}(1 - \sum_{l=1}^{k} \beta_l) + \sum_{j=l}^{J}\sum_{i=1}^{N_j}\sum_{k'=k+1}^{K} \delta(r_{ji,l}=1, z_{ji,k'}=1)\Bigg]. \tag{3.14}$$

3. The update of the indicator vector $\mathbf{r}_{ji}$, for $j = 1, \ldots, J$ and $i = 1, \ldots, N_j$, is completed by generating samples from a multinomial distribution with entries

$$Pr(r_{ji,l} = 1|\cdots) \propto \tilde{w}_{l-1} \prod_{m=l}^{j-1}(1-\tilde{w}_m) \prod_{k=1}^{K} \left\{\tilde{\pi}_{lk} \prod_{q=1}^{k-1}(1-\tilde{\pi}_{lq}) \cdot Pr(x_{ji}|\boldsymbol{\theta}_k^*)\right\}^{z_{ji,k}}, \ \ l = 1, ..., j \tag{3.15}$$

with $Pr(x_{ji}|\boldsymbol{\theta}_k^*)$ the likelihood of x_{ji} given allocation to the kth atom, $\boldsymbol{\theta}_{ji} = \boldsymbol{\theta}_k^*$. The posterior probability $Pr(r_{ji,l} = 1)$ is normalized so $\sum_{l=1}^{j} Pr(r_{ji,l} = 1) = 1$.

4. The sampling of the indicator vector $\mathbf{z}_{ji}$, for $j = 1, \ldots, J$ and $i = 1, \ldots, N_j$, is also generated from a multinomial distribution with entries specified as

$$Pr(z_{ji,k} = 1|\cdots) \propto \prod_{l=1}^{j} \left\{\tilde{\pi}_{lk} \prod_{k'=1}^{k-1}(1 - \tilde{\pi}_{lk'}) \cdot Pr(x_{ji}|\boldsymbol{\theta}_k^*)\right\}^{r_{ji,l}}, \ \ k = 1, ..., K. \tag{3.16}$$

Other unknowns, including $\{\boldsymbol{\theta}_k^*\}_{k=1}^{K}$, $\{\tilde{\beta}_k\}_{k=1}^{K-1}$ and precision parameters γ, $\boldsymbol{\alpha}_0$, are updated using standard Gibbs steps. The component parameters $\boldsymbol{\theta}_k^*$ for $k = 1, \ldots, K$ are considered for different model forms depending on the specific applications. The Gibbs sampling algorithm was tested carefully under different initializations and the diagnostic method in [RL92] is used to demonstrate rapid convergence and good mixing (for the results considered, convergence based on this method was observed for a burn-in of 200 samples, followed by a subsequent 4000 samples).

3.2 Music Analysis with Dynamic HDP

The analysis of music is of interest to music theorists, for aiding in music teaching, for analysis of human perception of sounds [Tem08], and for design of music search and organization tools [NPCD08]. An example of the use of Bayesian techniques for analyzing music may be found in the work [Tem07]. However, this work is generally assumed that the user has access to MIDI files (musical instrument digital interface), which means that the analyst knows exactly what notes are sounding when. We are interested in processing the acoustic waveform directly; while the techniques developed here are of interest for music, they are also applicable for analysis of general acoustic waveforms. For example, a related problem which may be addressed using the proposed approach is the segmentation of audio waveforms for automatic speech and speaker recognition (*e.g.*, for labeling different speakers in a teleconference [FSJW08b]).

As motivation we start by considering a well-known musical piece: "A Day in the Life" from the Beatles' album *Sgt. Peppers Lonely Hearts Club Band*. The piece is 5 minutes and 33 seconds long, and the entire audio waveform is plotted in Figure 3.4. To process these data, the acoustic signal was sampled at 22.05 KHz and divided into 50 ms contiguous frames. Mel frequency cepstral coefficients (MFCCs) [Log00] were extracted from each frame, these being effective for representing perceptually

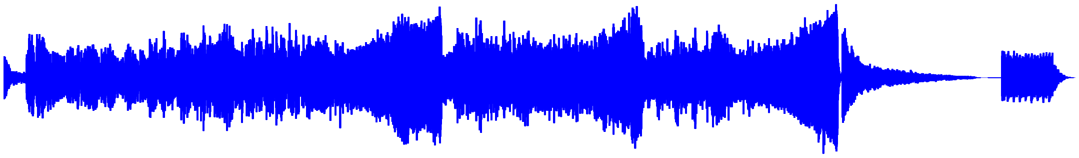


FIGURE 3.4: The audio waveform of the Beatles' music.

important parts of the spectral envelope of audio signals [JCMJ06]. The MFCC features are linked to spectral characteristics of the signal over the 50 ms window, and this mapping yields a 40-dimensional vector of real numbers for each frame. Therefore, after the MFCC analysis the music is converted to a sequence of 40-dimensional real vectors.

The details of the model follow below, and here we only seek to demonstrate our objective. Specifically, Figure 3.5 shows a segmentation of the audio waveform, where the indices on the figure correspond to data subsequences; each subsequence is defined by a set of 75 consecutive 50 ms frames. The results in Figure 3.5 quantify how interrelated any one subsequence of the music is to all others. We observe that the music is decomposed into clear contiguous segments of various lengths, and segment repetitions are evident. This Beatles' song is a relatively simple example, for the piece has many distinct sections (vocals, along with clearly distinct instrumental parts). A music-theoretic analysis of the results in Figure 3.5 indicates that the segmentation correctly captures the structure of the music. In the detailed results presented below, we consider much "harder" examples. Specifically, we consider classical piano music for which there are no vocals, and for which distinct instruments are not present (there is a lack of timbral variety, which makes this a more difficult

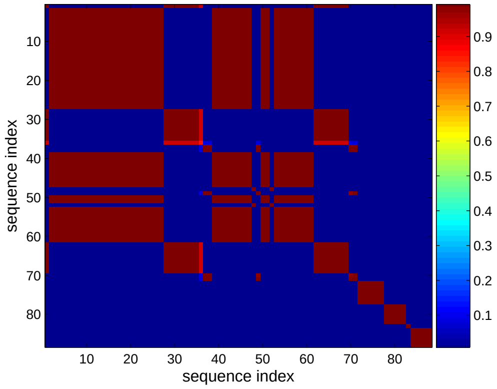


FIGURE 3.5: Segmentation of the audio waveform in Figure 3.4.

challenge). We also provide a detailed examination of the quality of the inferred music segmentation, based on music-theoretic analysis.

A typical goal of the music analysis is to segment a given piece, with the objective of inferring interrelationships among motive and themes within the music. Paiement et al. [PGBE07] proposed a generative model for rhythms based on the distributions of distances between subsequences; to annotate the changes in mixed music, Plotz et al. [PFH$^+$06] used stochastic models based on the Snip-Snap approach, by evaluating the Snip model for the Snap window at every position within the music. However, these methods are either based on one specific factor (rhythm) of music [PGBE07] or need prior knowledge of the music's segmentation [PFH$^+$06]. Recently, a hidden Markov model (HMM) [Rab89] was used to model monophonic music by assuming all the subsequences are drawn i.i.d. from one HMM [Rap99]; alternatively, an HMM mixture [QPC07] was applied to model the variable time-evolving properties of music, within a semiparametric Bayesian setting. In both of these HMM music models the music was divided into subsequences, with an HMM employed to represent each subsequence; such an approach does not account for the expected statistical relationships between temporally proximate subsequences. By considering one piece of music as a whole (avoiding subsequences), an infinite HMM (iHMM) [TJBB06, NPCD08] was proposed to automatically learn the model structure with countably infinite states. While the iHMM is an attractive model, it has limitations for the music modeling and segmentation of interest here, with this discussed further below.

As indicated at the beginning, a given piece of music is mapped to a sequence of 40-dimensional real vectors via MFCC feature extraction. The MFCCs are the most widely employed features for processing audio signals, particularly in speech processing. To simplify the HMM mixture models employed here, each 40-dimensional real vector is quantized via vector quantization (VQ) [GG92], and here the codebook is of dimension $M = 16$. For example, after VQ, the continuous waveform in Figure

3.4 is mapped to the sequence of codes depicted in Figure 3.6; it is a sequence of this type that we wish to analyze.

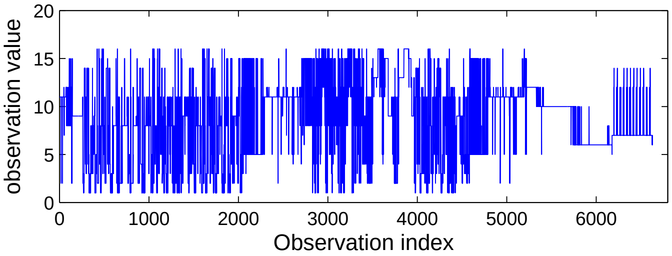


FIGURE 3.6: Sequence of code indices for the waveform in Figure 3.4, using a codebook of dimension $M = 16$.

To model the whole music piece with one HMM [Rap99], one may divide the sequence into a series of subsequences $\{\mathbf{x}_j\}_{j=1}^{J}$, with $\mathbf{x}_j = \{x_{jt}\}_{t=1}^{T}$ and $x_{jt} \in \{1, ..., M\}$. However, rather than employing a single HMM for a given piece, which is clearly overly simplistic, we allow the music dynamics to vary with time by letting

$$\mathbf{x}_j \sim F(\boldsymbol{\theta}_j), \quad j = 1, \ldots, J, \tag{3.17}$$

which denotes that the subsequence $\mathbf{x}_j$ is drawn from an HMM with parameters $\boldsymbol{\theta}_j$. In order to accommodate dependence across the subsequences, we can potentially let $\boldsymbol{\theta}_j \sim G_j$, and G_j has been defined in Eq. (3.6) with the dHDP introduced in Section 3.1.1.

Accordingly, the dHDP-HMM can be summarized as follows:

$$\boldsymbol{\theta}_j \sim G_j, \quad G_j = \sum_{k=1}^{\infty} p_{jk}\delta_{\boldsymbol{\theta}_k^*}, \quad \boldsymbol{\theta}_k^* \sim H, \tag{3.18}$$

where the subsequence-specific mixture distribution G_j has weights that vary with j, represented as $\mathbf{p}_j$. Based on the dependent relation induced in Eq. (3.6), we have an explicit form for each $\{\boldsymbol{p}_j\}_{j=1}^J$ in (3.18):

$$\boldsymbol{p}_j = \sum_{l=1}^{j} w_{jl} \boldsymbol{\pi}_l. \tag{3.19}$$

Including the same atoms for all j allows for repetition in the music structure across subsequences, with the varying weights allowing substantial flexibility.

As in the work [QPC07], the component parameters $\mathbf{A}_k^*$, $\mathbf{B}_k^*$ and $\boldsymbol{\pi}_k^*$ are assumed to be *a priori* independent, with the base measure having a product form with Dirichlet components for each of the probability vectors. The update equations for the components' posterior are presented in Appendix B. For each subsequence, there are still two indicator vectors $\mathbf{r}_j$ and $\mathbf{z}_j$ to denote the selection of the mixture distribution and component respectively. Since the indicator vector $\mathbf{z}_j$, for $j = 1, \ldots, J$, represents the membership of sharing across all the subsequences, we use this information to segment the music, by assuming that the subsequences possessing the same membership should be grouped together. In order to overcome the issue of label switching that exists in Gibbs sampling, we use the similarity measure $E(\mathbf{z}'\mathbf{z})$ instead of the membership $\mathbf{z}$ in the results. Here $E(\mathbf{z}'\mathbf{z})$ is approximated by averaging the quantity $\mathbf{z}'\mathbf{z}$ from multiple iterations, and in each iteration $\mathbf{z}_j'\mathbf{z}_{j'}$ measures the sharing degree of $\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_{j'}$ by integrating out the index of atoms. Related clustering representations of nonparametric models have been considered in the work [MS02].

3.3 Experiment Results

To apply the dHDP-HMM to music data, we first complete the prior specification by choosing hyperparameter values. In particular, the prior for $\tilde{\mathbf{w}}$ is chosen to encourage the groups to be shared; consequently, we set the prior $\prod_{j=1}^{J-1} Be(\tilde{w}_j; a_w, b_w)$ with $a_w = 1$ and $b_w = 5$. Since the precision parameters γ and $\boldsymbol{\alpha}_0$ control the prior distribution on the number of clusters, the hyper-parameter values should be chosen carefully. Here we set $Ga(1,1)$ for γ and each component of $\boldsymbol{\alpha}_0$. Meanwhile, we set the truncation level for DP at $K = 40$.

3.3.1 Statistical Analysis of Music Piece

The music considered below are from particular audio recordings, and may be listened to online[1]. We first consider the movement ("Largo - Allegro") from the Beethoven's Sonata No. 17, Op. 31, No. 2 (the "Tempest"). The audio waveform of this piano music is shown in Figure 3.7. The music is divided into contiguous 100 ms frames, and for each frame the quantized MFCC features are represented by one code from a codebook of size $M = 16$. Each subsequence is of length 60 (corresponding to 6 seconds in total), and for the Beethoven piece considered here there are 83 contiguous subsequences ($J = 83$). The lengths of the subsequences were carefully chosen based on consultation with a music theorist to be short enough to capture meaningful fine-scale segmentation of the piece. To represent the time dependence inferred by the model, the posterior of indicator $\mathbf{r}$ is plotted in Figure 3.8 (a) to show the mixture-distribution sharing relationship across different subsequences. Figure 3.8 (b) shows the similarity measures $E(\mathbf{z}'\mathbf{z})$ across each pair of subsequences, in which the higher value represents larger probability of the two corresponding subsequences being shared; here $\mathbf{z}$ (see (3.16)) is a column vector containing one at the position to be associated with the component occupied at the current iteration and zeros otherwise.

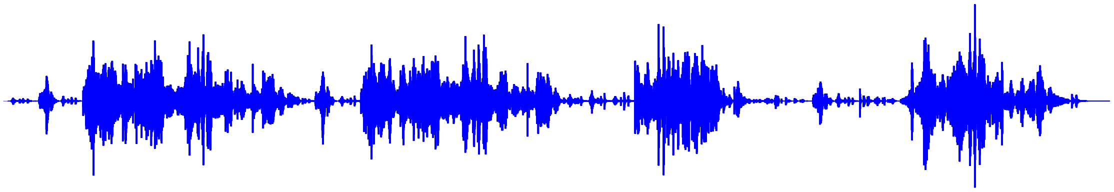
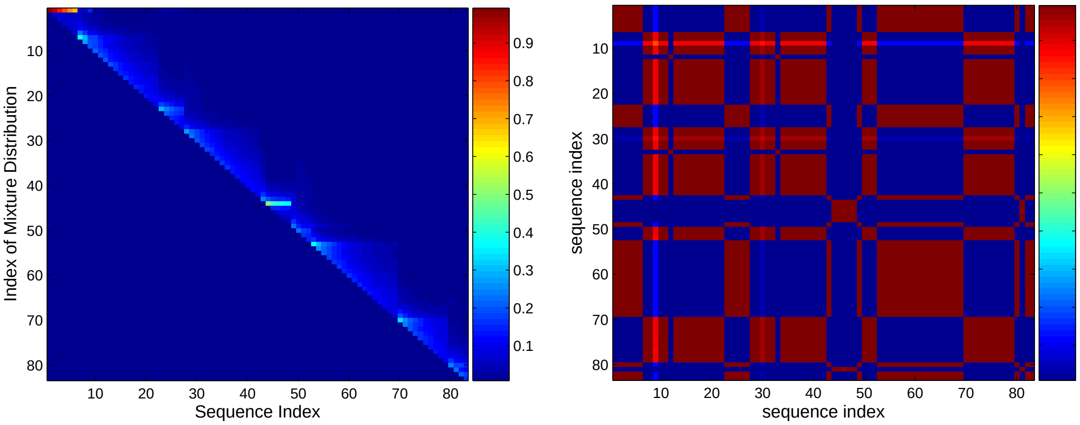


FIGURE 3.7: Audio waveform of the first movement of Op. 31, No. 2.

[1] http://www.last.fm/

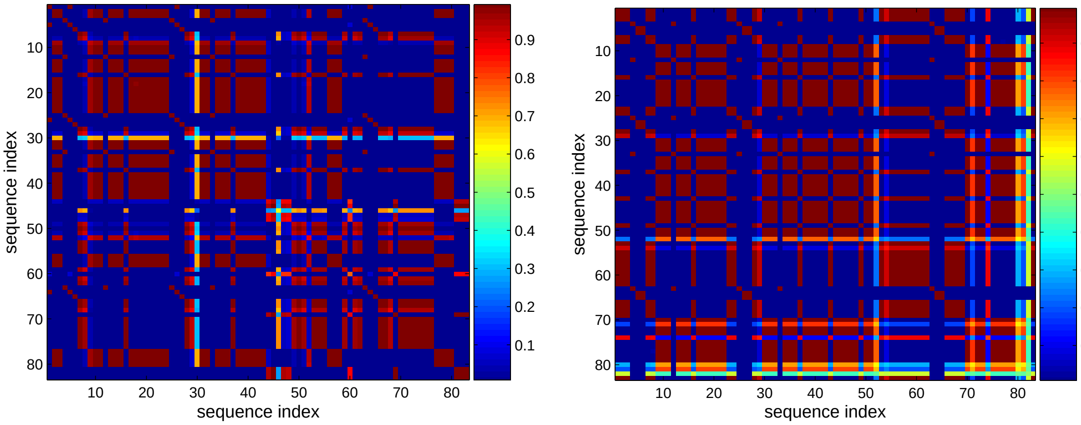


(a) (b)

Figure 3.8: Results of dHDP HMM modeling for the Sonata No. 17. (a) The posterior distribution of indicator variable $\mathbf{r}$. (b) The similarity matrix $E[\mathbf{z}'\mathbf{z}]$.

For comparison, we now analyze the same music using a DP-HMM [QPC07], HDP-HMM [TJBB06] and an iHMM [BGR02, TJBB06]. In the DP-HMM, we use the model in (2.4), with $F(\cdot)$ corresponding to an HMM with the same number of states as used in the dHDP; this model yields an HMM mixture model across the music subsequences, and the subsequence order is exchangeable. However, the long time dependence for the music's coherence is not considered in the components sharing mechanism. For the DP-HMM, we used the same specification of the base measure, H, as in the dHDP-HMM. A Gamma prior Ga(1,1) is employed as the hyper-prior for the precision parameter α in (2.4) and the truncation level is also set to 40. The DP-HMM inference was performed with MCMC sampling [QPC07]. We also consider a limiting case of the dHDP-HMM, for which all innovation weights are zero, with this referred to as an HDP-HMM, with inference performed as in the dHDP, simply with the weights removed. As formulated, the HDP-HMM yields a posterior estimate on the HMM parameters (atoms) for each subsequence, while the DP-HMM yields a posterior estimate on the HMM parameters (atoms) across all of the subsequences. Thus, the HDP-HMM yields an HMM mixture model for each subsequence, and the mixture atoms are shared across all subsequences; for the DP-HMM a single HMM mixture model is learned across all subsequences.

As in Figure 3.8, we plot the similarity measures $E(\mathbf{z}'\mathbf{z})$ across each pair of subsequences for DP-HMM in Figure 3.9 (a) and also show the same measure from HDP-HMM in Figure 3.9 (b), in which the dynamic structure is removed from dHDP; other variables have the same definition as inferred via the DP-HMM and HDP-HMM. Compared with the result of dHDP in Figure 3.8 (b), we observe a clear difference: although the DP-HMM can also tell the repetitive patterns occurring before the 42^{th} subsequence, the HMM components shared during the whole piece jump from one to the other between the successive subsequences, which makes it difficult to segment the music and understand the development of the piece (*e.g.*, the slow solo part between the 53^{th} and 69^{th} subsequences is segmented into many small pieces in DP-HMM); similar performance is also achieved in the results of HDP-HMM [Figure 3.9 (b)] and the music's coherence structure is not observed in such modelings.

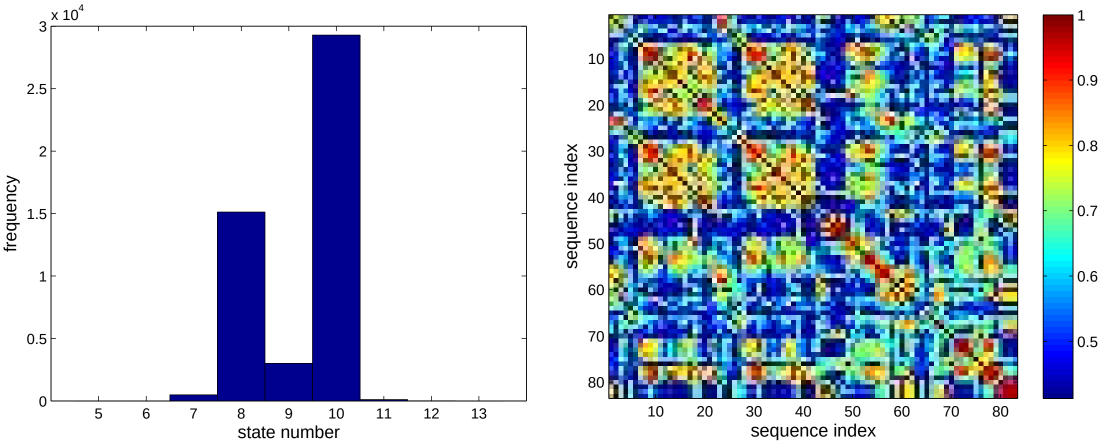


(a)

(b)

FIGURE 3.9: Results of DP-HMM and HDP-HMM mixture modeling for the Sonata No.17. (a) The similarity matrix $E(\mathbf{z}'\mathbf{z})$ from DP-HMM result. (b) The similarity matrix $E(\mathbf{z}'\mathbf{z})$ from HDP-HMM result.

Additionally, we also compare the dHDP HMM with segmentation results produced by the iHMM [BGR02, TJBB06]. With the iHMM, the music is treated as one long sequence (all the subsequences are concatenated together sequentially) and a single HMM with an "infinite" set of states is inferred; in practice, a finite set of states is inferred as probable, as quantified in the state-number posterior. For the piece of music under consideration, the posterior on the number of states across the entire piece is as depicted in Figure 3.10 (a). The inference was performed using MCMC, as in [TJBB06], with hyper-parameters consistent with the models discussed above.



(a) (b)

FIGURE 3.10: Analysis results for the piano music based on the iHMM. (a) Posterior distribution of state number. (b) Approximate similarity matrix by KL-divergence.

With the MCMC, we have a state estimation of each observation (codeword, for our discrete-observation model). For each of the subsequences considered by the other models, we employ the posterior on the state distribution to compute the Kullback-Leibler (KL) divergence between every pair of subsequences. Since the KL divergence is not symmetric, we define the distance between two state distribution as $D = \frac{1}{2}\{E(D_{KL}(P_1||P_2)) + E(D_{KL}(P_2||P_1))\}$. Based on the collected samples, we use the averaged KL divergence to measure the similarity between any two subsequences and plot it in Figure 3.10(b). Although such a KL-divergence matrix is a little noisy, we observe a similar time-evolving sharing existing between adjacent subsequences, as inferred by the dHDP. This is because the iHMM also characterizes the music's coherence since all of the sequential information is contained in one HMM. However, the inference of this relationship requires a postprocessing step with the iHMM, while the dHDP infers these relationships as a direct aspect of the inference, also yielding "cleaner" results.

3.3.2 Model Quality Relative to Music Theory

The results of our computational analysis are compared with segmentations performed by a composer, musician, and professor of music [2]. This music analysis is based upon reading the musical notes as well as listening to the piece being played. The music-theoretic analysis was performed *independent* of the numerical analysis (performed by the other authors), and then the relationship between the two analysis was assessed by the professor of music. We did *not* perform a numerical analysis and then subsequently interpret the results; the music analysis and numerical analysis were performed independently, and subsequently compared. The results of this comparison are discussed below.

For this comparison, the temporal resolution of the numerical analysis is increased; in the example presented below 15 discrete observations represent one second of music, each subsequence is again of length $T = 60$ (4 second subsequences), and for the Beethoven piece we now have $J = 125$ contiguous frames. All other parameters are unchanged. In Figure 3.11 it is observed that the model does a good job of segmenting the large sectional divisions found in sonata form: exposition, exposition repeat, development, and recapitulation (discussed further below). Along the top of the figure, we note a parallel row of circles (white) and ellipses (yellow); these

[2] All the music interpretation in Chapter 3 was done by Professor Scott Lindroth from Department of Music, Duke University.

correspond to the "Largo" sections and are extracted well. The first two components of "Largo" (white circles) are exact repeats of the music, and this is reflected in the segmentation. Note that the yellow ellipses are still part of "Largo", but they are slightly distinct from the yellow circles at left; this is due to the introduction of new key areas extended by recitative passages. The row of white squares correspond to the "main theme" ("Allegro"), and these are segmented properly. The parallel row of blue rectangles corresponds to the second key area, which abruptly changes the rhythmic and melodic texture of the music. Note that the first two of these (centered about approximately sequences 30 and 58) are exact repeats. The third appearance of this passage is in a different key, which is supported by the graph showing slightly lower similarity.

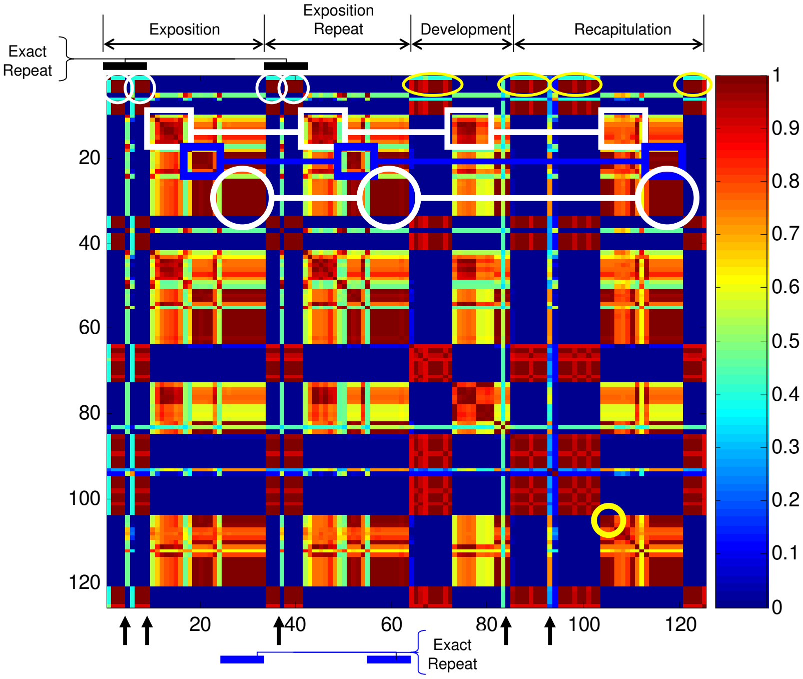


FIGURE 3.11: Annotated $E(\mathbf{z}'\mathbf{z})$ for the Beethoven sonata. The description of the annotations are provided in the text. The numbers along the vertical and horizontal axes correspond to the sequence index, and the color bar quantifies the similarity between any two segments.

The row of three circles parallel to approximately sequence 30 corresponds to another sudden change in texture characterized by melodic neighbor tone motion emphasizing Neapolitan harmony (A-natural moving to B^b), followed by a harmonic sequence. The rightmost circle, in the recapitulation, is in a different key and consequently emphasizes neighbor motion on D-natural and E^b, and is still found to be similar to the earlier two appearances.

We also note that the A-natural / B^b neighbor motion is similar to subsequences near subsequence 20, and this may be because subsequence 20 also has strong neighbor tone motion (E-natural to F-natural) in the left-hand accompaniment.

Finally, the bottom-right circle in Figure 9 identifies unique material that replaces the recapitulation of the main theme ("Allegro"), and its similarity to the main theme (around sequence 16) moves lower. The arrows at the bottom of Figure 19 identify "Allegro" interjections in the Largo passages, not all of which are in the same key.

3.3.3 Analysis of Mozart Piece

The above example examined the performance of the dHDP model relative to other competing statistical approaches, and to nonstatistical (more traditional) analysis performed by the third author. Having established the utility of dHDP relative to the other statistical approaches, we now only consider dHDP for the next example: Mozart K. 333, Movement 1 (sampled with each frame 50 ms long, yielding for this case $J = 139$ subsequences). This is again entirely a piano piece. We now provide a more complete sense of how the traditional musical analysis was performed, and provide a fuller examination of dHDP relative to such analysis, for the Mozart piece.

Above we considered the first movement of Beethoven's Sonata No. 17, Op. 31, No. 2 (the "Tempest"), and below is considered the first movement of Mozart's Sonata K. 333. Classical sonata movements have a consistent approach to the presentation and repetition of themes as well as a clear tonal structure. The first movement of K. 333 by Mozart frequently appears in music anthologies used in undergraduate courses in music theory and history and often held up as a typical example of sonata form [Bur03]. The first movement of Op. 31, No. 2 by Beethoven is an example of the composer's self-conscious effort to expand the technical and expressive vocabulary of sonata form, and the music shows a remarkable interplay of convention and innovation.

A classical sonata movement is a ternary form consisting of an Exposition (usually repeated), a Development, and a Recapitulation. The Exposition is subdivided into distinct subsections: a first theme in the tonic key, a second theme in the key of the dominant (or relative major for minor key sonata movements), and a closing theme in the dominant (or relative major). A transition between the first and second themes modulates from the tonic key to the dominant. The closing theme may be followed by a coda to conclude the Exposition in the key of the dominant.

The Development typically draws on fragments from the Exposition themes for melodic material. These are recombined to construct sequential patterns which modulate freely (observing the conventions of Classical harmony). It is not unusual for entirely new themes to be introduced. In most cases, the Development ends with a retransition which extends dominant harmony in preparation for the return to tonic harmony. The Recapitulation presents the first theme again in the tonic key, a modified transition, the second theme, now in the tonic key instead of the dominant, followed by the closing theme and coda, all in the tonic key.

This patterned circulation of themes and key areas gives sonata form a pleasing predictability-the knowledgeable listener can anticipate what is going to happen next-as well as a built-in tension that results from a tonal structure that establishes the tonic key, departs for the dominant key, moves through passages of harmonic instability, and finally releases harmonic tension by a return to the tonic key.

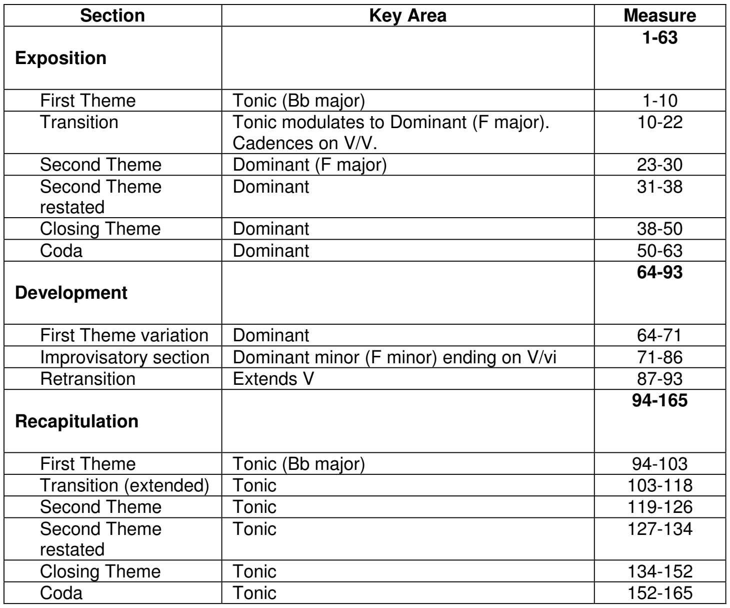
Traditional Analysis of K. 333 by W.A. Mozart

K. 333 closely follows the template described above. Measures 1 – 10 present the first theme in the tonic key (Bb major). Measures 10 – 22 present the transition based on the first theme, but modified in such a way that the music cadences on the dominant. The second theme appears in the key of the dominant (F major) in mm. 23 – 30 and is restated in mm. 31 – 38. The closing theme follows in mm. 38 – 50, and mm. 50 – 63 comprise a coda which brings the Exposition to a conclusion in F major, the dominant key.

As is typical for a Mozart a sonata, the first and second themes are clearly distinguished from each other. The first theme is harmonically stable and maintains a consistent texture of melody and accompaniment. In contrast, the second theme juxtaposes several short thematic ideas that introduce dynamic and textural changes, chromatic inflections, rhythmic syncopations, and virtuosic passage work. The closing theme is distinguished from both the first and second themes by an Alberti bass accompaniment in sixteenth notes and faster melodic motion.

The Development begins in m. 64 with a variation of the first theme in the key of F major. The theme cadences deceptively in the key of F minor in m. 71, which begins a new section cast in an improvisatory character that ends with a chromatic descent to the dominant of the submediant (V/vi) in m. 81. The retransition in mm. 87 – 93 abruptly introduces dominant harmony and prepares for the return to the tonic key of Bb major.

The Recapitulation begins in m. 94 with a restatement the first theme in the tonic key. Measures 94–103 are an exact restatement of mm. 1–10. The transition follows in mm. 104 – 118. Like the corresponding passage in the Exposition, this passage is based on the first theme, however, it is extended to accommodate a harmonic excursion that cadences on the dominant. The second theme, also in the tonic key, follows in mm. 119 – 134. Aside from the transposition to the tonic key, this passage is nearly an exact repetition of mm. 23 – 38, with the restatement of the second theme played an octave higher in mm. 127 – 134. The closing theme in mm. 134 – 152 is now stated in the tonic key as expected, however, like the transition, it is extended by a harmonic sequence in mm. 143 – 146 and by the insertion of entirely new material in mm. 147 – 151. The coda in measures 152-165 is an exact repetition of mm. 50 – 63, except now transposed to the tonic key. The thematic/harmonic analysis is summarized in Figure 3.12.



Section	Key Area	Measure
Exposition		**1-63**
First Theme	Tonic (Bb major)	1-10
Transition	Tonic modulates to Dominant (F major). Cadences on V/V.	10-22
Second Theme	Dominant (F major)	23-30
Second Theme restated	Dominant	31-38
Closing Theme	Dominant	38-50
Coda	Dominant	50-63
Development		**64-93**
First Theme variation	Dominant	64-71
Improvisatory section	Dominant minor (F minor) ending on V/vi	71-86
Retransition	Extends V	87-93
Recapitulation		**94-165**
First Theme	Tonic (Bb major)	94-103
Transition (extended)	Tonic	103-118
Second Theme	Tonic	119-126
Second Theme restated	Tonic	127-134
Closing Theme	Tonic	134-152
Coda	Tonic	152-165

FIGURE 3.12: Summary of the traditional musical analysis of Sonata for Piano, K. 333, First Movement.

Tracking themes and key areas is rather simple in K. 333 since it closely adheres to the sonata template. Such an exercise is a typical assignment in an undergraduate music theory course. A more subtle analysis focuses on contrapuntal design as well as on the use of chromaticism at different structural levels. For example, it is entirely characteristic of Haydn, Mozart, and Beethoven to introduce chromatic melodic embellishments as local events which later serve as a contrapuntal or voice leading "scaffold" projected over many measures, or even over entire sections of a piece. This is seldom audible, even to a sophisticated listener, however, it is a central aspect of compositional technique in the Classical period, one that creates a sense of continuous, organic development across sectional divisions. K. 333 offers an excellent example of this technique[3].

The closing theme and coda in the Exposition introduce a chromatic melodic descent based the pitches F-E-Eb-D. The use of chromaticism for local color has been a prominent feature of the second theme, and thus the appearance of the chromatic descent in the closing theme does not seem unusual. The chromatic figure can be seen and heard in mm. $46-47$, $50-51$, $54-55$, and $59-62$. The same chromatic descent appears twice in the Development section, the first time projected over mm. $64-68$, and the second time projected over mm. $71-81$, the improvisatory passage in the key of F minor. Thus, what appeared to be entirely new music in the Development (mm. 71 ff.) is actually derived from the chromatic melodic descent introduced in the Exposition. This is a perfect example of unity underlying variety.

A successful dHDP analysis of K. 333 should segment the music in a way that corresponds to sectional divisions of sonata form. Since our performance repeats the Exposition, we would expect dHDP to show strong similarity between the two statements of the first theme, transition, second theme, closing theme, and coda. The Recapitulation presents an interesting challenge. While all thematic materials from the Exposition appear in the Recapitulation, everything from the transition to the end is stated in the tonic key instead of the dominant key. In other words, the Recapitulation has strong melodic similarity to the Exposition, but the notes are different. The Development offers another challenge. While this section begins

[3] Analysis of contrapuntal and chromatic details at multiple structural levels was developed by the German theorist, Heinrich Schenker (1868-1935)

with a variation of the first theme, the improvisation that follows is (seemingly) entirely new music. If anything, dHDP analysis might show the similarity of the improvisation to the closing theme because both passages make use of Alberti bass figuration in sixteenth notes. A truly remarkable analysis would catch the projection of chromatic details over long passages in the Development section.

Segmentation by dHDP Analysis of K. 333

Before beginning the analysis of Figure 3.13, it should be emphasized that precise linkage between music-theoretic analysis and statistical analysis is difficult, since for the latter the music is divided into a series of contiguous 4-second blocks (these blocks do not in general line up precisely with music-theoretic segments in the music). This makes detailed analysis of some passages more difficult, particularly when several small segments appear in close succession. Having said this, dHDP analysis segments the music appropriately (based on the expert judgment).

Considering the annotations in Figure 3.13, the vertical arrow at the bottom identify unaccompanied melodic transitions in the right hand or sudden changes to soft dynamics, which are generally distinguished by the dHDP. The first row of white circles (near the top) correspond to the beginning of the second theme, characterized by the distinctive chordal gesture in the key of the dominant, and this decomposition or relationship appears to be accurate. We note that the third appearance of this gesture in the recapitulation is in a different key, and the similarity is correspondingly lower. An example of an "error" is manifested in the row of white rectangles. These correspond to the closing theme, and the left two rectangles (high correlation between each) are correct, but the right rectangle does not have a corresponding high correlation inside; it is therefore not recognized in the recap, when it appears in a different key (tonic). The results in Figure 3.13 show a repeated high degree of similarity that is characteristic of Mozart piano sonatas; the consistent

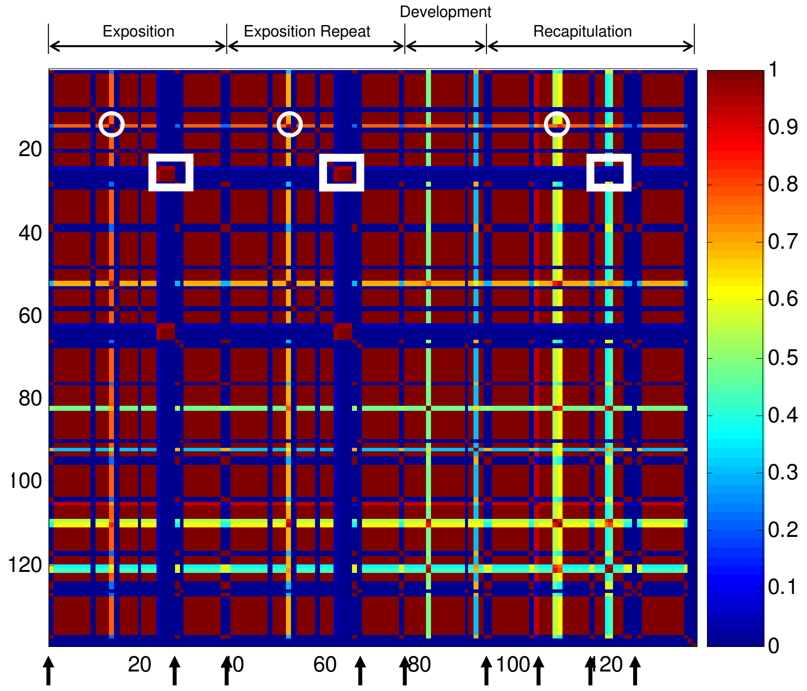


FIGURE 3.13: Annotated $E(\mathbf{z}'\mathbf{z})$ for the Mozart. The description of the annotations are provided in the text. The numbers along the vertical and horizontal axes correspond to the sequence index, and the color bar quantifies the similarity between any two segments.

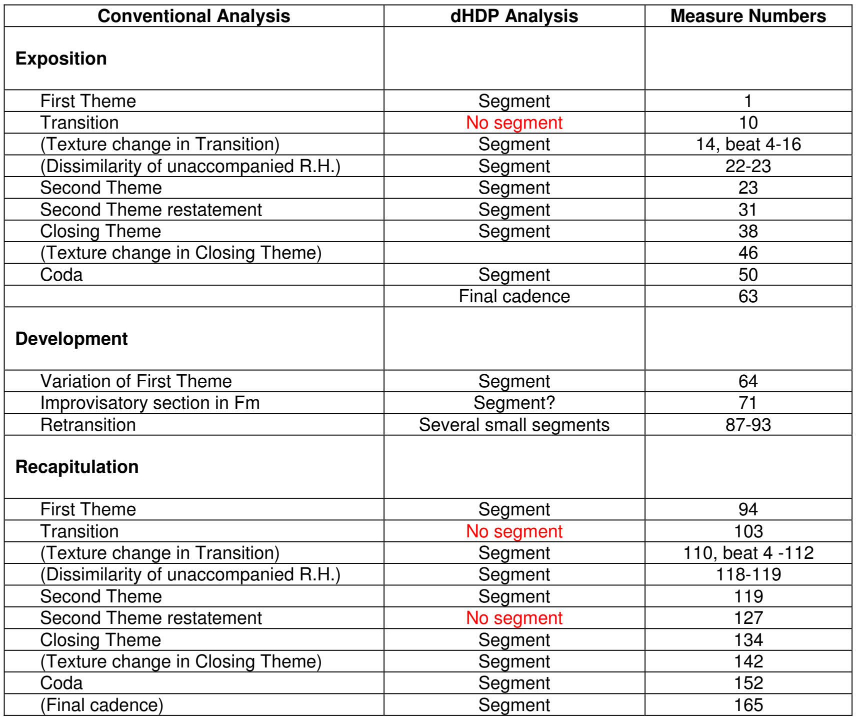
musical structure is occasionally permeated by exquisite details, such as a phase transition (these, again, identified by the arrows at the bottom).

The large sectional divisions between the Exposition, Development, and Recapitulation are easily seen in Figure 3.13. This figure also marks the beginnings of the first theme, second theme, closing theme, and coda within the Exposition. The beginning of the transition section is not distinguished from the first theme in Figure 3.13, despite the clear cadence that separates the first theme and transition. On the other hand, dHDP isolates a brief passage that occurs in the middle of the transition (m. 14, beat 4 - m. 16). This passage is characterized by a sudden change in dynamics and register. Other examples of local segmentation appear at the end of the transition and the beginning of the second theme (mm. $22-23$), when the right hand is unaccompanied by the left. Here Figure 3.13 shows a prominent orange band denoting less similarity with the music immediately preceding and following this passage, which is entirely consistent with the musical texture. The figure marks the restatement of the second theme (m. 31) and isolates the final measures of the coda when the musical texture thins out at the Exposition cadence. The sudden change of texture and dynamics within the closing theme (mm. $46 - 48$) is clearly separated from the main part of the closing theme in the figure. Even smaller segments comprising a few notes are marked. These segments isolate moments between phrases when the right hand plays quietly, unaccompanied by the left hand. The dHDP analysis of the Exposition repeat precisely replicates the segmentation described above.

The Development is represented as a single block, though the beginning of the improvisatory section in F minor (m. 71) appears to be marked by a prominent green band, indicating less similarity with the music immediately preceding and following this moment. Figure 3.13 marks the retransition with several small segments, however, the resolution of the figure makes it difficult to correlate these segments with particular moments in the music. Figure 3.13 clearly marks the Recapitulation with its return to the first theme in the tonic key. As before, the beginning of the transition goes unnoticed, however, dHDP again segments the transition passage associated with a sudden change in register and dynamics (mm. 110, beat $4 - 112$).

The end of the transition and beginning of the second theme (mm. $118 - 119$) is marked by a prominent orange/yellow band (Figure 3.13) indicating less similarity, just as was seen at the same moment in the Exposition (mm. $22-23$). The figure does not mark the restatement of the second them as it did in the Exposition, however, this may be a consequence of misalignment between the music playback and the analysis, as discussed above. The closing theme is segmented appropriately, and the sudden change of texture and dynamics in mm. $142 - 146$ is segmented apart from the rest of the closing theme, just as we saw in mm. 46 – 48 in the Exposition. Note that Figure 3.13 clearly shows this passage has been extended to five measures in the Recapitulation compared to three measures in the Exposition. The figure segments the coda in the same way we saw in the Exposition, including its isolation of the final cadence.

In sum, dHDP analysis has segmented the music remarkably well. Parallel passages which appear throughout the movement are represented the same way each time they occur. Even the omissions are consistent, such as the lack of segmentation of the transition from the first theme. The results are summarized in Figure 3.14.



Conventional Analysis	dHDP Analysis	Measure Numbers
Exposition		
First Theme	Segment	1
Transition	No segment	10
(Texture change in Transition)	Segment	14, beat 4-16
(Dissimilarity of unaccompanied R.H.)	Segment	22-23
Second Theme	Segment	23
Second Theme restatement	Segment	31
Closing Theme	Segment	38
(Texture change in Closing Theme)		46
Coda	Segment	50
	Final cadence	63
Development		
Variation of First Theme	Segment	64
Improvisatory section in Fm	Segment?	71
Retransition	Several small segments	87-93
Recapitulation		
First Theme	Segment	94
Transition	No segment	103
(Texture change in Transition)	Segment	110, beat 4 -112
(Dissimilarity of unaccompanied R.H.)	Segment	118-119
Second Theme	Segment	119
Second Theme restatement	No segment	127
Closing Theme	Segment	134
(Texture change in Closing Theme)	Segment	142
Coda	Segment	152
(Final cadence)	Segment	165

FIGURE 3.14: Summary of the dHDP analysis of Sonata for Piano, K. 333, First Movement.

Quality of Similarity Defined by dHDP Analysis of K. 333

The dHDP analysis shows a high degree of similarity of most thematic materials in the movement. For example, the first theme, transition, second theme, and coda are all marked with the highest degree of similarity to each other across the entire movement. The dHDP analysis does not appear to recognize the differences in note successions in these passages.

Figure 3.13 does indicate moments of dissimilarity. For example, the closing theme (beginning in m. 38) is marked as dissimilar from anything else in the movement. Recall that the closing theme introduced a new Alberti bass accompaniment in sixteenth notes which helps set this music apart. However, the reappearance of the closing theme in the Recapitulation is not represented as similar to the closing theme in the Exposition. Perhaps the transposition of the closing theme to the tonic key in the Recapitulation obscures the similarity, but this does not explain why the closing theme in the Recapitulation is marked as highly similar to the first and second themes, transition, and development throughout the rest of the movement, no matter what key they are in.

Several incidental details are marked with a high degree of similarity to each other while being dissimilar to the rest of the movement. These are normally moments when the music suddenly becomes quiet or features isolated groups of notes played by the right hand without accompaniment. Examples of this can be seen along the horizontal axis at the very top of Figure 3.13 and include the pickups to m. 1, m. 46, the pickups to m. 64, the pickups to m. 94, mm. 118 – 119, m. 130, mm. 142 – 146, and the final cadence in m. 165.

Finally, we observe that dHDP analysis is not suitable for revealing the projection of chromatic details at a larger structural level. We note, however, that dHDP analysis did mark the first appearance of the descending chromatic melodic figure in mm. $46 - 48$ of the closing theme.

From these results we may suppose that similarity in dHDP analysis is more strongly associated with dynamics, texture, and register than with melody and harmony. This raises an important point. While dynamics can be specified in the musical score, it is up to the musician to interpret these markings in performance. It is possible that dHDP analysis would represent another interpretation of the same piece differently. For brevity, we only provide the detailed music-theoretic analysis, with comparison to dHDP, for the Mozart piece. However, the same detailed analysis was used to yield the conclusions above with respect to the Beethoven piece. That analysis is provided online as Supplement 2. We reiterate that the music-theoretic analysis of the type summarized in Figure 3.12 was performed *independent* of the statistical analysis, with comparisons performed subsequently.

3.4 Summary

The dynamic hierarchical Dirichlet process (dHDP) has been developed for analysis of sequential data, with a focus on analysis of audio data from music. The framework assumes a parametric representation $F(\boldsymbol{\theta})$ to characterize the statistics of the data observed at a single point in time. The parameters $\boldsymbol{\theta}$ associated with a given point in time are assumed to be drawn from a mixture model, with in general an infinite number of atoms, analogous to the Dirichlet process. The mixture models at time $t-1$ and time t are interrelated statistically. The model is linked to the hierarchical Dirichlet process [TJBB06] in the sense that the initial mixture model and the subsequent time-dependent mixtures are drawn from the same discrete distribution. This implies that the underlying atoms in the $\boldsymbol{\theta}$ space associated with the aforementioned mixtures are the same, and what is changing with time are the mixture weights. The model has the following characteristics: (i) with inferred probabilities, the underlying parameters associated with data at adjacent times are the same; and (ii) since the same underlying atoms are used in the mixtures at all times, it is possible that the same atoms may be used at temporally distant time, allowing the capture of repeated patterns in temporal data. The underlying sharing properties (correlations) between observations at adjacent times have also been derived. Inference has been performed in an MCMC setting.

Examples have been presented on three musical pieces: a relatively simple piece from the Beatles, as well as two more complicated classical pieces. The classical pieces are more difficult to analyze because there are no vocals, and a single instrument is generally used, and therefore the segmentation of such data is more subtle. The results of the classical-piece segmentations have been analyzed for their connection to music analysis. In this connection it is felt that the results are promising. While there were mistakes in the analysis of the Beethoven and Mozart pieces considered, there is a great deal of accuracy as well. The results clearly reveal meaningful characteristics about Beethoven and Mozart.

The dHDP analysis effectively segments the two classical compositions by Mozart and Beethoven at both the large-scale and local levels. Segmentation appears to be related to musical dynamics, texture, and register. The dHDP analysis of similarity is far more successful in the Beethoven sonata than in the work by Mozart. It may be that the greater variety of musical textures, dynamics, and registral placement in Op. 31, No. 2 yield more gradations of similarity in the graph. The dHDP analysis makes several plausible similarity connections, though there are inconsistencies as well. The greatest deficiency in the dHDP analysis of similarity is the apparent inability to track note successions (*i.e.*, themes) and key areas as a basis for comparison.

Despite these shortcomings, dHDP analysis is instructive for musicians, perhaps especially so for composers (these are observations of the third author, who is a composer and musician). In K. 333, Mozart articulates form through themes and tonal structure. Beethoven articulates form in Op. 31, No. 2 through themes that are linked to emphatic gestures, as well as through a detailed tonal design. This is not to say that one is better than the other. There are works by Beethoven that may result in findings that are similar to K. 333, and Mozart has composed works that may result in findings that are as varied as the results of Op. 31, No. 2. Nonetheless, dHDP analysis of K. 333 and Op. 31, No. 2 illustrates general tendencies of the two composers that are commonly acknowledged by musicians and audiences alike.

Concerning future research, for large data sets the MCMC inference engine employed here may not be computationally tractable. The graphical form of the dHDP is applicable to more-approximate inference engines, such as a variational Bayesian (VB) analysis [BJ04]. We intend to examine VB inference in future studies, and to examine its relative advantages in computational efficiency compared to its inference accuracy (relative to MCMC). Additionally, our model was motivated by a stick-breaking construction of DP; however, it is also of interest to consider a Chinese restaurant/franchise representation [TJBB06], which may have advantages for interpretation and inference.

4

Nonparametric Modeling of Spatial Dependence

Spatial data analysis has been widely used in many scientific and engineering applications, such as geometric field exploration, social network data mining, wireless system simulations, remote sensing and imaging systems etc. For these problems, the data are collected at specific locations and the sampling measurements depend on their spatial origins. For example, the global precipitation displays a dynamic distribution as a function of geographic locations; the quality of domestic water varies across different regions; even the prevalent topics discussed within a social network may also exhibit spatial dependence within a local neighborhood. As the new technologies invented for global positioning systems, large spatial data with accurate geocoding of locations become available. This allows researchers to exploit the data to provide a qualitative spatial analysis or prediction.

To solve specific problems, spatial information will be considered in different ways. For prediction of local precipitation, the geographic coordinates may be used as predictors in a regression; For a spatial clustering problem, like image segmentation, the position of an object usually provides a prior information for the spatial partition; For remote sensing applications, multiple categories of targets need to be discriminated by establishing a set of local "experts systems" with spatial adaptivity. Underlying these different problems, there are at least two typical issues requires to discuss: how do we infer the spatial dependence underlying the data as the spatial locations are provided? And how many local regions should we consider for generating good estimation if the regional statistics are unknown in advance? To answer these questions, we think of using Bayesian nonparametric models to deal with the challenges in the same framework.

4.1 Extension from Temporal to Spatial Modeling

In the previous sections, probabilistic models for temporal data analysis have been studied. It makes one naturally think to extend a time-evolving statistical model to solve spatially dependent problems. With the data space changed from one dimension to two, we can hardly find a natural order to align the observations, so that a linear dynamic system with a Markov chain cannot be directly applied on modeling spatial dependence. In addition, for a 2-D space, the number of data points might be nonlinearly increased, sometimes leading to a huge computation cost and slow computation speed. Due to the potentially existed challenges, spatial modeling has been treated as an important studying field in statistics, and some traditional Bayesian techniques have been widely used.

One of the most popular methods is the Gaussian process [RW06], which establishes a correlation associated with data via placing a prior on the covariance structures. The spatial effect is customarily modeled as a mean-zero stationary Gaussian process (GP) [GKM05], composed of a set of jointly Gaussian random variables. Two proximate variables are expected to have a high correlation than those to be spatially distant. Assume we have a collection of data points $\{\mathbf{x}_i, \mathbf{s}_i\}_{i=1}^{N}$: $\mathbf{x}_i$ represents the observation in the feature space and $\mathbf{s}_i$ indicates the 2-D location coordinates.

The spatial dependence can be represented in the covariance function as follows:

$$Cov(\mathbf{x}_i, \mathbf{x}_j) = \alpha \text{exp}\big(-\frac{1}{2}\sum_{d=1}^{2} k_d (s_{id} - s_{jd})^2 \big), \tag{4.1}$$

in which α is an overall scaler to control the covariance value and k_d is the length scale to adjust the kernel smoothness. Given these parameters, we can calculate the correlation of any two data points in the 2-D surface and estimate the incoming data. One of the related work is the Kriging [Ste99], a class of geostatistical techniques for interpolation of spatial data. To explore the uncertainty of estimation at unsampled points, Gaussian processes are also extended for active data mining as spatially localized priors [RBkT+05, JG09]. With multiple outputs at the same region, Boyle [BF05] proposed dependent Gaussian processes to learn the spatial translations between outputs. High computational cost of Gaussian process modeling has also been tackled with sparse factorization [LI09] or projection to a lower dimensional subspace [BGFS08].

Another statistical model, Markov random field (MRF) [KS80], is also widely used in spatial modeling, especially for Bayesian segmentation. Markov random field is an n-dimensional random process defined on a discrete lattice. In a 2-D setting, the full conditional distribution of one random variable depends only on its neighborhood, defined as a set of its proximate sites. The objective with a Markov random field model is to minimize the sum of the deviation cost function and a penalty function that grows with the distance between the values of related pairs [BGFS08], hence introducing a localized dependent structure to model a prior belief about the contiguity of image features [LJ01, DC04, KP06]. In recent years, some scholars also considered a combination of MRF with nonparametric techniques [OB06, SYL+09], allowing the model with an adaptive complexity.

The main disadvantage of the Gaussian- and MRF-based methods is that the optimization of the parameters involves an expensive computation cost. This makes it difficult to directly apply the algorithms on large scale problems. Another potential issue for the Gaussian process is that the normal assumption may not always be true if the data itself shows a nonstationary property [KMH05], while the Markov random field needs to specify the neighborhood in advance. Due to the limitations of the both the two methods, we seek some more flexible ways for modeling spatially dependent data, with the following two models as starting motivations.

4.2 Generalized Spatial Dirichlet Process Model

Generalized spatial Dirichlet process [DGG07] is proposed for capturing residual spatial association of point-referenced spatial data $Y_D \equiv \{Y(s) : s \in D\}$, where s represents the locations that observations Y are collected. Although the model is established in the framework of Gaussian process, it generates a nonstationary, non-Gaussian random process, allowing different data sites to choose different random surfaces. This property leads to a more generalized form compared with the spatial Dirichlet process (SDP) proposed by Gelfand [GKM05]. The SDP model arises as a probability weighted collection of random surfaces, with each random surface being a realization from a base measure G_0. Assume G_0 is a Gaussian process and each realization from G_0 is represented as $\theta_{l,D} = \{\theta_l(s) : s \in D\}$ with $l = 1, \ldots, \infty$ being the index of the random surfaces. Based on the stick-breaking construction, the resulting distribution G for Y_D is

$$G = \sum_{l=1}^{\infty} w_l \delta_{\theta_{l,D}}, \quad w_l = z_l \prod_{r=1}^{l-1} (1 - z_r), \quad z_r \sim Be(1, \nu). \tag{4.2}$$

The marginal distribution at each site still comes from a Dirichlet process. Since the SDP is essentially a Dirichlet process defined on a space of surfaces, its realizations are discrete probability measures with countable support with probability one [GKM05]. Although the model specifies multiple random surfaces, sharing the same set of mixture weights still encourages all of the sites assigned to the same layer. Considering a more general form, both of the mixture components and distribution weights have the spatially dependent property, so that different sites choose the random surfaces according to their own selecting weights while the spatial dependence is still preserved.

Similar as the SDP, the generalized spatial Dirichlet process still draws each random surface $\{\theta_{l,D}\}_{l=1}^{\infty}$ from a stationary Gaussian process. Its one realization G on the space D allows the surface selection to vary with the choice of locations, and is still a discrete probability measure with countable support with probability one. Accordingly, for any set of locations $(s_1, \ldots, s_n) \in D$ and collection $\{A_1, \ldots, A_n\}$ in $\mathcal{B}(\mathbb{R})$,

$$P\{Y(s_1) \in A_1, \ldots, Y(s_n) \in A_n\} = \sum_{i_1=1}^{\infty} \ldots \sum_{i_n=1}^{\infty} p_{i_1,\ldots,i_n} \delta_{\theta^*_{i_1}(s_1)}(A_1) \ldots \delta_{\theta_{i_n} A^*(s_n)}(A_n), \tag{4.3}$$

satisfying the condition that $\sum_{i_1=1}^{\infty} \ldots \sum_{i_n=1}^{\infty} p_{i_1,\ldots,i_n} = 1$ with any $p_{i_1,\ldots,i_n} \geq 0$. In addition, the weights defined on the infinite dimensional simplex should also satisfy a spatial continuity property: as $s \to s_0$, the joint probability $p_{i_1,i_2} = p\{Y(s) = \theta^*_{i_1}(s), Y(s_0) = \theta^*_{i_2}(s_0)\}$ converges to the marginal probability $p_{i_2} = p\big(Y(s_0) = \theta^*_{i_2}(s_0)\big)$ when $i_1 = i_2$, and to 0 otherwise. For the joint probability of n ($n \in N$) locations, a similar conclusion can be reached.

Based on the above assumption, the mixture distribution weights are generated via a set of gating variables, as represented as follows:

$$\begin{aligned} p_{i_1,\ldots,i_n} = p\big[&z_1(s_1) < 0, \ldots, z_{i_1-1}(s_1) < 0, z_{i_1}(s_1) \geq 0; \\ &z_1(s_2) < 0, \ldots, z_{i_2-1}(s_2) < 0, z_{i_2}(s_2) \geq 0; \\ &\vdots \\ &z_1(s_n) < 0, \ldots, z_{i_n-1}(s_n) < 0, z_{i_n}(s_n) \geq 0\big], \end{aligned} \tag{4.4}$$

where $z_l(s_1), \ldots, z_l(s_n)$ are a countable collection of indicator variables drawn from a Gaussian process associated with the lth random surface; for a specific location s, $z_1(s) < 0, \ldots, z_{i-1}(s) < 0, z_i(s) \geq 0$ jointly indicate the site $Y(s) = \theta_i^*(s)$. Within this framework, the marginal distribution of the generalized spatial Dirichlet process is still a Dirichlet process :$G(s) = \sum_{l=1}^{\infty} p_l(s)\delta_{\theta_l^*(s)}$, where $p_l(s)$ is defined as $p\big[z_1(s) < 0, \ldots, z_{l-1}(s) < 0, z_l(s) \geq 0\big]$. Although the model admits a flexible spatial dependence, the use of GPs presents computational challenges as applied to large scale problems.

4.3 Kernel Stick-Breaking Process

Kernel stick-breaking process (KSBP) is proposed for uncountable collections of dependent random probability measures [DP07]. It still considers an infinite number of atoms drawn from a base G_0 like the traditional DP, while incorporates predictors within the mixture weights via a bounded kernel multiplied by the beta weights. The KSBP imposes that clustering is more probable if two feature vectors are close in a prescribed (general) space, which may be associated explicitly with spatial position for spatially dependent data modeling. With the KSBP, rather than assuming exchangeable data, one realization of the random measures G becomes a function of spatial location:

$$\begin{aligned} G_{\mathbf{s}} &= \sum_{k=1}^{\infty} \pi_k(\mathbf{s}; V_k, \Gamma_k, \psi)\delta_{\boldsymbol{\theta}_k^*} \\ \pi_k(\mathbf{s}; V_k, \Gamma_k, \psi) &= V_k K(\mathbf{s}, \Gamma_k; \psi) \prod_{k'=1}^{k-1} \left[1 - V_{k'} K(\mathbf{s}, \Gamma_{k'}; \psi)\right] \\ V_k \sim \mathrm{Beta}(1, \alpha_0), \quad & \boldsymbol{\theta}_k^* \sim G_0, \quad \Gamma_k \sim H_0 \end{aligned} \tag{4.5}$$

where $K(\mathbf{s}, \Gamma_k; \psi)$ represents a kernel distance between the feature-vector spatial coordinate $\mathbf{s}$ and a local basis location Γ_k associated with the kth stick; ψ is the kernel width to adjust the distance at an appropriate scale.

Note that the measure $G_{\mathbf{s}}$ drawn from the KSBP turns into a function of location, with each basis probability weight π_k introduced for different location Γ_k. As the basis location Γ_k is close to the predictor location $\mathbf{s}$, the bounded kernel function associated with Γ_k yields a larger value than others, favoring a high probability to select the atom with the same index k; however, this is only true for those atoms also having smaller index due to the property of the stick-breaking process. Meanwhile, the KSBP includes the Dirichlet process as a special case, when $K(\mathbf{s}, \Gamma_k; \psi) = 1$ for all (s, Γ_k). An advantage of the KSBP formulation is that many of the tools developed for exchangeable stick-breaking processes can be applied with minimum modification [DP07] and the simple form of the model makes it straightforward to formulate the inference. Although the kernel function introduced a spatial dependence in KSBP to encourage the proximate feature vectors to be clustered together, it does not yield a spatial contiguity for the resulting clustering, as the image segmentation result shown in [AWS$^+$08].

Motivated by the above discussions and considering a specific application, image segmentation, we still use a kernel function to measure the spatial distance like KSBP, but rebuild the spatial dependence within the generalized spatial Dirichlet process (GSDP) as another form. The new model, logistic stick-breaking process (LSBP), can be applied on a large scale problem with a relatively fast variational inference; meanwhile it also shows a good performance compared with other state-of-art methods.

5

Logistic Stick-Breaking Process

5.1 Introduction

One is often interested in clustering data that have associated spatial or temporal coordinates. This problem is relevant in a diverse set of applications, such as climatology, ecology, environmental health, real estate marketing, and image analysis [BCG03]. The available spatial or temporal information may be exploited to help infer patterns, clusters or segments in the data. To simplify the exposition, in the following discussion we focus on exploiting *spatial* information, although when presenting results we also consider *temporal* data [FSJW08a].

There have been numerous techniques developed to cluster data, although most of these do not explicitly exploit appended spatial information. One class of state-of-the-art methods employs graphical techniques, such as normalized cuts [SM00, FH04] and extensions [ZK04]. These approaches regard the two-dimensional (2D) data as an undirected weighted graph, and the segmentation is equivalent to finding the minimum cut of the graph, minimizing the between-group disassociation while maximizing the within-group association [SM00]. Such graph-theoretic methods have attractive computational speed, but do not provide a statistical inference (measure of confidence), and often one must pre-define the total number of segments/clusters. Further, such graphical techniques are not readily extended to the joint analysis of multiple spatially dependent data sets, with this of interest for the simultaneous analysis of multiple images.

To consider clustering in a nonparametric Bayesian manner, the Dirichlet process (DP) [BM73] has been employed widely [Ant74, EW95, Ras00, BGR02]. The assumption within the DP that the data are exchangeable is generally inappropriate when one wishes to impose knowledge of spatial information (in which each $\mathbf{y}_n$ has an associated spatial location). For example, the data may be represented as $\{\mathbf{y}_n, \mathbf{s}_n\}_{n=1}^{N}$, in which $\mathbf{y}_n$ is again the feature vector and $\mathbf{s}_n$ represents the spatial location of $\mathbf{y}_n$. Provided with such spatial information, one may wish to explicitly impose the belief that proximate data are more likely to be clustered together.

The spatial location $\mathbf{s}_n$ may be readily considered as an *appended* feature, and the modified feature vectors (data) may then be analyzed via traditional clustering algorithms, like those discussed above. For example, the spatial coordinate has been considered explicitly in recent topic models [CFF07, WG07, GWP08] when applied in image analysis. These previous studies seek to cluster visual words, with such clustering encouraged if the features are spatially proximate. However, these methods may produce spurious clusters that are introduced to better characterize the spatial data likelihood instead of the likelihood of the features conditionally on spatial location [PD09]. In addition, such approaches require a model for the spatial locations, which is not statistically coherent as these locations are typically fixed by design, and there may be additional computational burden for this extra component.

To address these challenges, and impose spatial information more explicitly, researchers have recently modified the DP construction to manifest spatial-location dependent stick weights. The work of Duan *et al.* [DGG07] recently introduced a framework in terms of a hierarchy of Gaussian processes (see section 4.2), in which the spatially dependent construction is obtained by thresholding K latent Gaussian processes (GPs); while this is a powerful construction, the use of GPs presents computational challenges [SJ08]. To simplify the model structure, the Dirichlet labeling process [PGG09] has been proposed, in which one thresholds only one latent Gaussian process to regulate spatial dependence. However, the model inference, performed with Markov chain Monte Carlo (MCMC), is inefficient for many large-scale applications. Similar issues are also true for work that has combined the Dirichlet process with a Markov random field (MRF) constraint [OB08]. As an alternative to the above approaches, a kernel stick-breaking process (KSBP) has been proposed [DP07] and discussed in section 4.3. As demonstrated when presenting results, the KSBP generally does not yield smooth segments with sharp boundaries.

Instead of thresholding K latent Gaussian processes [DGG07] to assign a feature vector to a particular parameter, we introduce a novel non-parametric spatially dependent prior, called the logistic stick-breaking process (LSBP), to impose that it is probable that proximate feature vectors are assigned to the same parameter. The new model is constructed based on a hierarchy of spatial logistic regressions, with sparseness-promoting priors on the regression coefficients. With this relatively simple model form, inference is performed efficiently with variational Bayesian analysis [Bea03], allowing consideration of large-scale problems. Further, for reasons discussed below, this model favors contiguous segments with sharp boundaries, of interest in many applications. The model developed in [CD09a], based on a *probit* stick-breaking process, is most closely related to the proposed framework; the relationships between LSBP and the model in [CD09a] are discussed in detail below.

5.2 Model Specifications

We first consider spatially constrained clustering for a single data set (task). Assume N sample points $\{\mathbf{D}_n\}_{n=1,N}$, where $\mathbf{D}_n = (\mathbf{y}_n, \mathbf{s}_n)$, with $\mathbf{y}_n$ representing the nth feature vector and $\mathbf{s}_n$ its associated spatial location. We draw a set of candidate model parameters, and the probability that a particular space-dependent data sample employs a particular model parameter is defined by a spatially-dependent stick-breaking process, represented by a kernel-based logistic-regression.

Assume an infinite set of model parameters $\{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty}$. Each observation $\mathbf{y}_n$ is drawn from a parametric distribution $F(\boldsymbol{\theta}_n)$, with $\boldsymbol{\theta}_n \in \{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty}$. To indicate which parameter in $\{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty}$ is associated with the nth sample, a set of indicator variables $\mathbf{Z}_n = \{z_{n1}, z_{n2}, \ldots, z_{n\infty}\}$ are introduced for each $\mathbf{D}_n$, and all the indicator variables are equal to zero or one. Given $\mathbf{Z}_n$, data $\mathbf{D}_n$ is associated with parameter $\boldsymbol{\theta}_k^*$ if $z_{nk} = 1$ and $z_{n\hat{k}} = 0$ for $\hat{k} < k$.

The $\mathbf{Z}_n$ are drawn from a spatially dependent density function, encouraging that proximate $\mathbf{D}_n$ will have similar $\mathbf{Z}_n$, thereby encouraging spatial contiguity. This may be viewed in terms of a spatially dependent stick-breaking process. Specifically, let $p_k(\mathbf{s}_n)$ define the probability that $z_{nk} = 1$, with $1 - p_k(\mathbf{s}_n)$ representing the probability that $z_{nk} = 0$; the spatial dependence of these density functions is made explicit via $\mathbf{s}_n$. The probability that the kth parameter is selected in the above model is $\pi_k(\mathbf{s}_n) = p_k(\mathbf{s}_n)\prod_{\hat{k}=1}^{k-1}[1 - p_{\hat{k}}(\mathbf{s}_n)]$, which is of the same form as a stick-breaking process [IJ01] but extends to a spatially dependent mixture model, represented as

$$G_{\mathbf{s}_n} = \sum_{k=1}^{\infty} \pi_k(\mathbf{s}_n)\delta_{\boldsymbol{\theta}_k^*}, \quad \pi_k(\mathbf{s}_n) = p_k(\mathbf{s}_n)\prod_{\hat{k}=1}^{k-1}[1 - p_{\hat{k}}(\mathbf{s}_n)]. \tag{5.1}$$

Here each $p_k(\mathbf{s}_n)$ is defined in terms of a logistic link function (other link functions may also be employed, such as a probit). Specifically, we consider N_c discrete spatial locations $\{\hat{\mathbf{s}}_i\}_{i=1}^{N_c}$ within the domain of the data (*e.g.*, the locations of the samples in $\mathbf{D}_n$). To allow the weights of the different mixture components to vary flexibly with spatial location, we propose a kernel logistic regression for each break of the stick, with

$$\log\left(\frac{p_k(\mathbf{s}_n)}{1-p_k(\mathbf{s}_n)}\right) = g_k(\mathbf{s}_n) = \sum_{i=1}^{N_c} w_{ki} K(\mathbf{s}_n, \hat{\mathbf{s}}_i; \psi_k) + w_{k0} \tag{5.2}$$

where $g_k(\mathbf{s}_n)$ is the linear predictor in the logistic regression model for the kth break and position $\mathbf{s}_n$, and

$$K(\mathbf{s}_n, \hat{\mathbf{s}}_i; \psi_k) = \exp\Big[-\frac{\|\mathbf{s}_n - \hat{\mathbf{s}}_i\|^2}{\psi_k}\Big] \tag{5.3}$$

is a Gaussian kernel measuring closeness of locations $\mathbf{s}_n$ and $\hat{\mathbf{s}}_i$, as in a radial basis function model (alternative kernel functions may be defined). The kernel basis coefficients are represented as $\mathbf{W}_k = [w_{k0}, w_{k1}, \ldots, w_{kN_c}]'$. A sparseness-promoting prior is chosen for the components of $\mathbf{W}_k$, such that only a relatively small set of w_{ki} will have non-zero (or significant) amplitudes; those spatial regions for which the associated amplitudes are non-zero correspond to regions for which a particular model parameter is expected to dominate in the segmentation (this is similar to the KSBP in (4.5), which also has spatially localized kernels). The indicator variables controlling allocation to components are then drawn from

$$z_{nk} \sim \text{Bernoulli}[\sigma\big(g_k(\mathbf{s}_n)\big)] \tag{5.4}$$

where $\sigma(g) = 1/[1+\exp(-g)]$ is the inverse of the logit link in (5.2).

There are many ways that such sparseness promotion may be constituted, and we have considered two. As one choice, one may employ a hierarchical Student-t prior as applied in the relevance vector machine [BT00, Tip01, BS03]:

$$w_{ki} \sim N(w_{ki}|0, \lambda_{ki}^{-1})\text{Gamma}(\lambda_{ki}|a_0, b_0) \tag{5.5}$$

where shrinkage is encouraged with $a_0 = b_0 = 10^{-6}$ [Tip01]. Alternatively, one may consider a "spike-and-slab" prior [IR05]. Specifically,

$$w_{ki} \sim \nu_k \mathcal{N}(0, \lambda_k^{-1}) + (1 - \nu_k)\delta_0, \quad \nu_k \sim \text{Beta}(\nu_k | c_0, d_0) \tag{5.6}$$

The expression δ_0 represents a unit point measure concentrated at zero. The parameters (c_0, d_0) are set such that ν_k is encouraged to be close to zero, enforcing sparseness in $\boldsymbol{w}_k$; the parameter λ_k is again drawn from a gamma prior, with hyperparameters set to allow a possibly large range in the non-zero values of w_{ki}, and therefore these are *not* set as in the Student-t representation. The advantage of the latter model is that it explicitly imposes that many of the components of $\boldsymbol{w}_k$ are exactly zero, while the Student-t construction imposes that many of the coefficients are close to zero. In our numerical experiments on waveform and image segmentation, we have employed the Student-t construction.

Note that parameter $\boldsymbol{\theta}_k^*$ is associated with an $\boldsymbol{s}$-dependent function $g_k(\boldsymbol{s})$, and there are $K - 1$ such functions. The model is constructed such that within a contiguous spatial/temporal region, a particular parameter $\boldsymbol{\theta}_k^*$ is selected, with these model parameters used to generate the observed data.

There are two key components of the LSBP construction: (*i*) sparseness promotion on the w_{ki}, and (*ii*) the use of a logistic link function to define space-dependent stick weights. As discussed further in section 5.3, these concepts are motivated by the idea of making a particular space-dependent LSBP stick weight $\pi_k(\boldsymbol{s}) = \sigma(g_k(\boldsymbol{s})) \prod_{k'<k}[1 - g_k'(\boldsymbol{s})]$ near one within a localized region in space (motivating the sparseness prior on the weights), while also yielding contiguous segments with sharp boundaries (manifested via the logistic).

It is desirable to allow flexibility in the kernel parameter ψ, as this will influence the size of segments that are encouraged (discussed further below). Hence, for each

k we draw

$$\psi_k = \psi^*_{r_k} \qquad r_k \sim \text{Mult}(1/\tau, \dots, 1/\tau) \tag{5.7}$$

with $\mathbf{\Psi}^* = \{\psi^*_j\}_{j=1}^{\tau}$ a library of possible kernel-size parameters; r_k is an index for the one non-zero component of a *single* draw from $\text{Mult}(1/\tau, \dots, 1/\tau)$. We employ a discrete dictionary of kernel sizes $\mathbf{\Psi}^*$ because there is not a conjugate prior for imposition of a continuous distribution of kernel parameters (this is discussed further in section 5.4). A draw from this hierarchical prior is denoted concisely as $G_{\mathbf{s}} \sim \text{LSBP}(H, a_0, b_0, \mathbf{\Psi}^*)$, where it is assumed that we are using the Student-t prior for weights $\{\boldsymbol{w}_k\}_{k=1,K-1}$, with a similar representation used for a spike-and-slab prior; note that $G_{\mathbf{s}}$ is defined simultaneously for *all* spatial locations. The model parameters $\{\boldsymbol{\theta}^*_k\}_{k=1}^{\infty}$ are assumed drawn from the measure H.

In practice we usually truncate the LSBP to K sticks, as in a truncated stick-breaking process [IJ01]. With a truncation level K specified, if $z_{nk} = 0$ for all $k = 1, \dots, K-1$, then $z_{nK} = 1$ so that $\boldsymbol{\theta}_n = \boldsymbol{\theta}^*_K$. Since we yield an approximation to the full posterior density function via variational Bayesian (VB) inference (as discussed in section 5.4), we may also view selection of K as a model-selection problem. The VB analysis yields an approximation to the marginal likelihood of the observed data, which can be used as a basis for model selection. When presenting results we consider simply setting K to a large value, or alternatively selecting K via model selection.

Figure 5.1 shows the graphical form of the model (using a Student-t sparseness prior), in which $\mathbf{\Psi}^*$ represents the discrete set of kernel-width candidates, ψ_k is the kernel width selected for the kth stick, and the prior H takes on different forms depending upon the application. In Figure 5.1 the $1/\tau$ emphasizes that the candidate kernel widths are selected with uniform probability over the τ candidates in $\mathbf{\Psi}^*$.

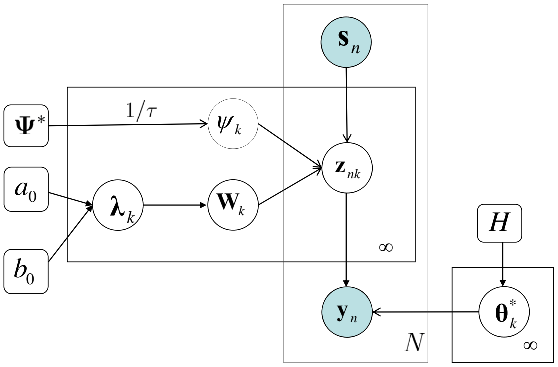


FIGURE 5.1: Graphical representation of the LSBP.

5.3 Discussion of LSBP Properties and Relationship to Other Models

The proposed model is motivated by the work in [SJ08], in which multiple draws from a Gaussian process (GP) are employed. Candidate model parameters are associated with each GP draw, and the GP draws serve to constitute a nonparametric gating network, associating particular model parameters with a given spatial position. In [SJ08] the spatial correlation associated with the GP draws induces spatially contiguous segments (a highly spatially correlated gating network), and this may be related to a spatially-dependent stick-breaking process. However, use of the GP produces computational challenges. The proposed LSBP model also manifests multiple space-dependent functions (here $g_k(\boldsymbol{s})$), with associated candidate model parameters $\{\boldsymbol{\theta}_k^*\}_{k=1,K}$. Further, we constitute a spatially dependent gating network that has a stick-breaking interpretation. However, a different and relatively simple procedure is proposed for favoring spatially contiguous segments with sharp boundaries.

At each location $\boldsymbol{s}$ we have a stick-breaking process, with the probability of selecting model parameters $\boldsymbol{\theta}_k^*$ defined as $\pi_k(\boldsymbol{s}) = \sigma(g_k(\boldsymbol{s}))\prod_{k'<k}[1-\sigma(g_{k'}(\boldsymbol{s}))]$. Recall that $g_k(\boldsymbol{s}) = \sum_{i=1}^{N_c} w_{ki}K(\boldsymbol{s}, \hat{\boldsymbol{s}}_i; \psi_k) + w_{k0}$, with sparseness favored for coefficients $\{w_{ik}\}_{i=0,N_c}$. Considering first $g_1(\boldsymbol{s})$, note that since most $\{w_{1i}\}_{i=1,N_c}$ are zero or near-zero, the bias w_{10} controls the stick weight $\pi_1(\boldsymbol{s})$ for all $\boldsymbol{s}$ sufficiently distant

from those locations $\hat{\boldsymbol{s}}_i$ with non-zero w_{1i}. Further, if $w_{1i} \gg 0$, $\sigma(g_1(\boldsymbol{s})) \approx 1$ for $\boldsymbol{s}$ in the "neighborhood" of the associated location $\hat{\boldsymbol{s}}_i$; the neighborhood *size* is defined by ψ_1. Hence, those $\{\hat{\boldsymbol{s}}_i\}_{i=1,N_c}$ with associated large $\{w_{1i}\}_{i=1,N_c}$ define localized regions as a function of $\boldsymbol{s}$ over which parameter $\boldsymbol{\theta}_1^*$ is highly probable, with locality defined by kernel scale parameter ψ_1. For those regions of $\boldsymbol{s}$ for which $\pi_1(\boldsymbol{s})$ is *not* near one, there is appreciable probability $1 - \pi_1(\boldsymbol{s})$ that model parameters $\{\boldsymbol{\theta}_k^*\}_{k=2,K}$ may be utilized.

Continuing the generative process, model parameters $\boldsymbol{\theta}_2^*$ are probable where $\pi_2(\boldsymbol{s}) = \sigma(g_2(\boldsymbol{s}))[1 - \pi_1(\boldsymbol{s})] \approx 1$. The latter occurs in the vicinity of those $\boldsymbol{s}$ that are distant from $\hat{\boldsymbol{s}}_i$ with large associated w_{1i} (*i.e.*, where $1 - \pi_1(\boldsymbol{s}) \approx 1$), while also being near $\hat{\boldsymbol{s}}_i$ with large w_{2i} (*i.e.*, where $\sigma(g_2(\boldsymbol{s})) \approx 1$). We again underscore that w_{20} impacts $\pi_2(\boldsymbol{s})$ for all $\boldsymbol{s}$.

This process continues for increasing k, and therefore it is probable that as k gets large all or almost all $\boldsymbol{s}$ will be associated with a large stick weight, or a large *cumulative* sum of stick weights, such that parameters $\boldsymbol{\theta}_k^*$ become improbable for large k and all $\boldsymbol{s}$.

Key characteristics of this construction are the clipping property of the logistic link function, and the associated fast rise of the logistic. The former imposes that there are contiguous regions (segments) over which the same model parameter has near-unity probability of being used. This encouraging of homogeneous segments is also complemented by sharp segment boundaries, manifested by the fast rise of the logistic. The aforementioned "clipping" property is clearly not distinct to logistic regression. It would apply as well to other binary response link functions, which can be any CDF for a continuous random variable. For example, probit links [CD09a] would have the same property, though the logistic has heavier tails than the probit so may have slightly different clipping properties. We have here selected the logistic link function for computational simplicity (it is widely used, for example, in the relevance vector machine [Tip01], and we borrow related technology). It is interesting to see how the segmentation realizations differ with the form of link function, with this to be considered in future research.

To give a more-detailed view of the generative process, we consider a one-dimensional example, which in section 5.5 will be related to a problem with real data. Specifically, consider a one-dimensional signal with 488 discrete sample points. In this illustrative example $N_c = 98$, defined by taking every fifth sample point for the underlying signal. We wish to examine the generative process of the LSBP prior, in the *absence* of data. For this illustration, it is therefore best to utilize the spike-and-slab construction, since without any data the Student-t construction will with high probability make all $w_{ki} \approx 0$ (when considering data, and evaluating the posterior, a small fraction of these coefficients are pulled away from zero, via the likelihood function, such that the model fits the data; we reconsider this in section 5.5). Further, again for illustrative purposes, we here treat $\{w_{k0}\}_{k=1,K}$ as drawn from a separate normal distribution, *not* from the spike-and-slab prior used for all other components of $\boldsymbol{w}_k$. This distinct handling of $\{w_{k0}\}_{k=1,K}$ has been found unnecessary when processing data, as the likelihood function again imposes constraints on $\{w_{k0}\}_{k=1,K}$. Hence this form of the spike-and-slab prior on $\boldsymbol{w}_k$ is simply employed to illuminate the characteristics of LSBP, with model implementation simplifying when considering data.

In Figure 5.2 we plot representative draws for $\boldsymbol{w}_k$, $g_k(\boldsymbol{s})$, $\sigma(g_k(\boldsymbol{s}))$ and $\pi_k(\boldsymbol{s})$, for the one-dimensional signal of interest. In this *illustrative* example each ν_k is drawn from Beta$(1, 10)$ to encourage sparseness, and those non-zero coefficients are drawn from $\mathcal{N}(0, \lambda)$, with λ fixed to correspond to a standard deviation of 15 (we could also draw each λ_k from a gamma distribution). Each bias term w_{k0} is here drawn i.i.d. from $\mathcal{N}(0, \lambda)$. We see from Figure 5.2 that the LSBP naturally favors localized segments that have near-unity probability of using the same model parameters. This

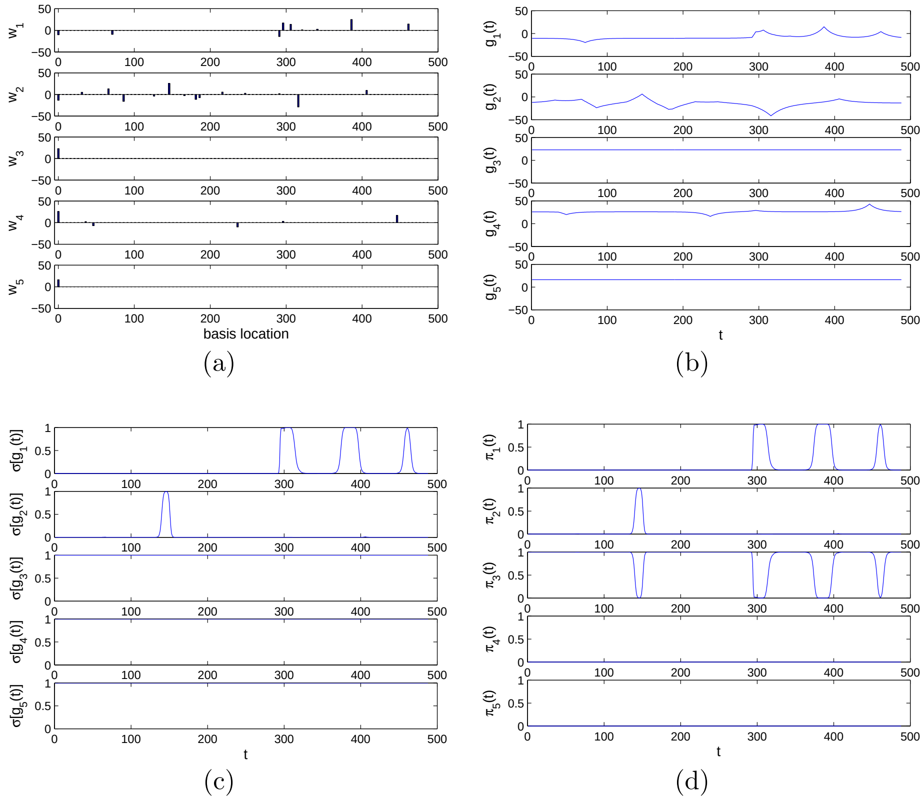


FIGURE 5.2: Example draw from a one-dimensional LSBP, using a spike-and-slab construction for model-parameter sparseness. (a) $\boldsymbol{w}_k$, (b) $g_k(t)$, (c) $\sigma_k(t)$, (d) $\pi_k(t)$

is a typical draw, where we note that for $k \geq 4$ the probability of θ_k^* being used is near zero. While Figure 5.2 represents a typical LSBP draw, one could also envision other less-desirable draws. For example, if $w_{10} \gg 0$ then $\pi_1(\boldsymbol{s}) \approx 1$ for all $\boldsymbol{s}$, implying that the parameters $\boldsymbol{\theta}_1^*$ is used for all $\boldsymbol{s}$ (essentially no segmentation). Other "pathological" draws may be envisioned. Therefore, we underscore that the data, via the likelihood function, clearly influences the posterior strongly, and the pathological draws supported by the prior in the absence of data are given negligible mass in the posterior.

As further examples, now for two-dimensional signals, Figure 5.3 considers example draws as a function of the kernel parameter ψ_k. These example draws were manifested via the same process used for the one-dimensional example in Figure 5.2,

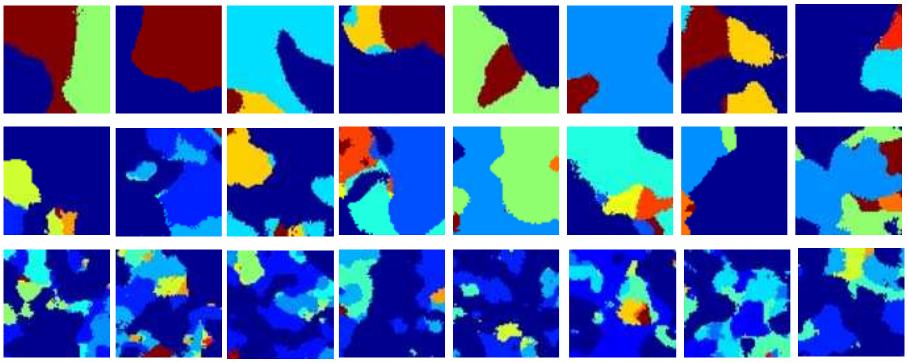


FIGURE 5.3: Samples drawn from the spatially dependent LSBP prior, for different (fixed) choices of kernel parameters ψ, applied for each k within the LSBP. In row 1 $\psi = 15$; in row 2 $\psi = 10$; and in row 3 $\psi = 5$. In these examples the spike-and-slab prior has been used to impose sparseness on the coefficients $\{\boldsymbol{w}_k\}_{k=1,K-1}$.

now extending $\boldsymbol{s}$ to two dimensions. Figure 5.3 also shows the dependence of the size of the segments on the kernel parameter ψ_k, which has motivated the learning of ψ_k in a data-dependent manner (based on a finite dictionary of kernel parameters $\boldsymbol{\Psi}^* = \{\psi_j^*\}_{j=1}^{\tau}$). The draws in Figure 5.3 are similar to those manifested by the GP-based construction in [SJ08], motivating the simple model developed here.

5.4 Model Inference

Markov chain Monte Carlo (MCMC) [GRS98] is widely used for performing inference with hierarchical models like LSBP. For example, many of the previous spatially-dependent mixtures have been analyzed using MCMC [DGG07, DP07, NG08, OB08]. The H-KSBP [AWS$^+$08] model is developed based on a Monte Carlo Variational Bayesian (MCVB) inference algorithm; however, nearly half of the model parameters still need to be estimated via a sampling technique. Although MCMC is an attractive method for such inference, the computational requirements may lead to implementation challenges for large-scale problems, and algorithm convergence is often difficult to diagnose.

The LSBP model proposed here may be readily implemented via MCMC sampling. However, motivated by the goal of fast and relatively accurate inference for large-scale problems, we consider variational Bayesian (VB) inference [Bea03].

5.4.1 Variational Bayesian analysis

Bayesian inference seeks to estimate the posterior distribution of the latent variables $\mathbf{\Phi}$, given the observed data $\mathbf{D}$:

$$p(\mathbf{\Phi}|\mathbf{D},\mathbf{\Upsilon}) = \frac{p(\mathbf{D}|\mathbf{\Phi},\mathbf{\Upsilon})p(\mathbf{\Phi}|\mathbf{\Upsilon})}{\int p(\mathbf{D}|\mathbf{\Phi},\mathbf{\Upsilon})p(\mathbf{\Phi}|\mathbf{\Upsilon})d\mathbf{\Phi}} \tag{5.8}$$

where the denominator $\int p(\mathbf{D}|\mathbf{\Phi},\mathbf{\Upsilon})p(\mathbf{\Phi}|\mathbf{\Upsilon})d\mathbf{\Phi} = p(\mathbf{D}|\mathbf{\Upsilon})$ is the model evidence (marginal likelihood); the vector $\mathbf{\Upsilon}$ denotes hyper-parameters within the prior for $\mathbf{\Phi}$. Variational Bayesian (VB) inference [Bea03] seeks a variational distribution $q(\mathbf{\Phi})$ to approximate the true posterior distribution of the latent variables $p(\mathbf{\Phi})$. The expression

$$\log p(\mathbf{D}|\mathbf{\Upsilon}) = L(q(\mathbf{\Phi})) + \mathrm{KL}(q(\mathbf{\Phi}) \parallel p(\mathbf{\Phi}|\mathbf{D},\mathbf{\Upsilon})), \tag{5.9}$$

with

$$L(q(\mathbf{\Phi})) = \int q(\mathbf{\Phi})\log\frac{p(\mathbf{D}|\mathbf{\Phi},\mathbf{\Upsilon})p(\mathbf{\Phi}|\mathbf{\Upsilon})}{q(\mathbf{\Phi})}d\mathbf{\Phi} \tag{5.10}$$

yielding a lower bound for $\log p(\mathbf{D}|\mathbf{\Upsilon})$ so that $\log p(\mathbf{D}|\mathbf{\Upsilon}) \geq L(q(\mathbf{\Phi}))$, since $\mathrm{KL}(q(\mathbf{\Phi}) \parallel p(\mathbf{\Phi}|\mathbf{D},\mathbf{\Upsilon})) \geq 0$. Accordingly, the goal of minimizing the KL divergence between the variational distribution and the true posterior reduces to adjusting $q(\mathbf{\Phi})$ to maximize (5.10).

Variational Bayesian inference [Bea03] assumes a factorized $q(\mathbf{\Phi})$, typically with the same form as employed in $p(\mathbf{\Phi}|\mathbf{D},\mathbf{\Upsilon})$. With such an assumption, the variational distributions can be updated iteratively to increase the lower bound. For the LSBP model applied to a single task, as introduced in section 5.2, we assume

$$q(\mathbf{\Phi}) = \prod_{k=1}^{K} q(\boldsymbol{\theta}_k) \prod_{k'=1}^{K-1} \left[q(\mathbf{w}_{k'}) q(\boldsymbol{\lambda}_{k'}) \prod_{n=1}^{N} q(z_{nk'}) \right], \tag{5.11}$$

where $q(\boldsymbol{\theta}_k)$ is defined by the specific application. In the audio-segmentation example considered below, the feature vector $\mathbf{y}_n$ may be assumed drawn from a multivariate normal distribution, and the K model parameters are means and precision matrices $\{\boldsymbol{\mu}_k^*, \mathbf{\Omega}_k^*\}_{k=1}^K$; accordingly $q(\boldsymbol{\theta}_k)$ is specified as a Normal-Wishart distribution (as is H), $N(\boldsymbol{\mu}_k|\tilde{\boldsymbol{\mu}}_k, \tilde{t}_k^{-1}{\mathbf{\Omega}_k}^{-1})\mathrm{Wi}(\mathbf{\Omega}_k|\tilde{\boldsymbol{V}}_k, \tilde{d}_k)$. For the rest of the model, $q(\mathbf{w}_{k'}) = \prod_{i=0}^{N_c} N(w_{k'i}|\tilde{m}_{k'i}, \tilde{\Gamma}_{k'i})$, $q(\boldsymbol{\lambda}_{k'}) = \prod_{i=0}^{N_c} Ga(\lambda_{k'i}|\tilde{a}_{k'i}, \tilde{b}_{k'i})$, and $q(z_{nk'})$ has a Bernoulli form $\rho_{nk'}^{z_{nk'}}(1-\rho_{nk'})^{1-z_{nk'}}$ with $\rho_{nk'} = \sigma(g_{k'}(n))$. The factorized representation for $q(\mathbf{\Phi})$ is a function of the hyper-parameters on each of the factors, with these hyper-parameters adjusted to minimize the aforementioned KL divergence.

By integrating over all the hidden variables and model parameters, the lower bound for the log model evidence

$$\begin{aligned}
\log p(\mathbf{D}|\mathbf{\Upsilon}) \quad & = \log \textstyle\int p\big(\mathbf{y}, \mathbf{s}, \boldsymbol{\theta}, \mathbf{W}, \boldsymbol{\lambda}, \mathbf{z}\big) \mathrm{d}\mathbf{\Phi} \\
& \geq \textstyle\int q(\boldsymbol{\theta}, \mathbf{W}, \boldsymbol{\lambda}, \mathbf{z}) \log \frac{p\big(\mathbf{y},\mathbf{s},\boldsymbol{\theta},\mathbf{W},\boldsymbol{\lambda},\mathbf{z}\big)}{q(\boldsymbol{\theta},\mathbf{W},\boldsymbol{\lambda},\mathbf{z})} \mathrm{d}\mathbf{\Phi} \\
& = \textstyle\int q(\boldsymbol{\theta}) q(\mathbf{W}) q(\boldsymbol{\lambda}) q(\mathbf{z}) \log \frac{p\big(\mathbf{y},\mathbf{s},\boldsymbol{\theta},\mathbf{W},\boldsymbol{\lambda},\mathbf{z}\big)}{q(\boldsymbol{\theta})q(\mathbf{W})q(\boldsymbol{\lambda})q(\mathbf{z})} \mathrm{d}\mathbf{\Phi} \\
& \equiv LB(q(\mathbf{\Phi})),
\end{aligned} \tag{5.12}$$

is a function of variational distributions $q(\mathbf{\Phi})$. The variational lower bound is optimized by iteratively taking derivatives with respect to the hyper-parameters in each $q(\cdot)$, and setting the result to zero while fixing the hyper-parameters of the other terms. Within each iteration, the lower bound is increased until the model converges.

The difficulty of applying VB inference for this model lies with the logistic-link function, which is not within the conjugate-exponential family. Based on bounding log convex functions, we use a variational bound for the logistic sigmoid function in the form [BS03]

$$\sigma(x) \geq \sigma(\eta)\exp\big(\frac{x-\eta}{2} - f(\eta)(x^2 - \eta^2)\big) \tag{5.13}$$

where $f(\eta) = \frac{\tanh(\eta/2)}{4\eta}$ and η is a variational parameter. An exact bound is achieved as $\eta = x$ or $\eta = -x$.

The detailed update equations are omitted for brevity, but are of the form employed in [Bea03] and [BS03]. Like other optimization algorithms, VB inference may converge to a local-optimal solution. However, such a problem can be alleviated by running the algorithm multiple times from different initializations (including varying the truncation level K, and for each case the atom parameters are initialized with k-mean clustering method [GG92] for a fast model convergence) and then using the solution that maximizes the variational model evidence.

5.4.2 Sampling the kernel width

As introduced in section 5.2, the kernel width ψ_k is inferred for each k. Due to the non-conjugacy of the sigmoid function, we cannot acquire a variational distribution for ψ_k. However, we can sample it from its posterior distribution by establishing a discrete set of potential kernel widths $\mathbf{\Psi}^* = \{\psi_j^*\}_{j=1}^{\tau}$, as discussed above. This resulting Monte Carlo Variational Bayesian (MCVB) algorithm combines both MCMC and VB inference, motivated by the Monte Carlo Expectation Maximization (MCEM) algorithm [WT90] and developed in [AWS$^+$08]. The intractable nodes within the graphical model are approximated with Monte Carlo samples from their conditional posterior distributions, and the lower bound of the log model evidence is generally with small fluctuations after the model converges [AWS$^+$08]. A detailed discussion on related treatments within variational Bayesian (VB) analysis may be found in [WB05] (see Section 6.3 of that paper).

Based on the variables $\mathbf{z}_n$, the cluster membership of each data $\mathbf{D}_n$ corresponding to different mixture components $\{\boldsymbol{\theta}_k^*\}_{k=1}^K$ can be specified as

$$\xi_{nk} = \prod_{k'=1}^{k-1} (1 - z_{nk'}) \cdot z_{nk}. \tag{5.14}$$

Based on the above assumptions, we observe that if $\xi_{nk} = 1$ and the other entries in $\xi_n = [\xi_{n1}, \ldots, \xi_{nK}]$ are equal to zero, then $\mathbf{y}_n$ is assigned to be drawn from $F(\boldsymbol{\theta}_k^*)$.

With the variables $\boldsymbol{\xi}$ introduced and a uniform prior $\mathbf{U}$ assumed on the kernel width $\{\psi_j^*\}_{j=1}^{\tau}$, the posterior distribution for each ψ_k is represented as

$$\begin{array}{l} p(\psi_k = \psi_j^* | \cdots) \propto U_j \cdot \exp\big\{ \sum_n < \xi_{nk} > \big[< \log\sigma\big(g_k^j(\mathbf{s}_n)\big) > \big] \big\} \cdot \\ \exp\big\{ \sum_n \sum_{l>k} < \xi_{nl} > \big[< \log\Big(1 - \sigma\big(g_k^j(\mathbf{s}_n)\big)\Big) > \big] \big\}, \end{array} \tag{5.15}$$

where U_j is the jth component of $\mathbf{U}$, $< \cdot >$ represents the expectation with the associated random variables, $g_k^j(\mathbf{s}_n) = \sum_{i=1}^{N_c} w_{ki} K(\mathbf{s}_n, \hat{\mathbf{s}}_i; \psi_j^*) + w_{k0}$ with $j = 1, \ldots, \tau$.

With the definition $\mathbf{x}_n^j = \Big[1, K(\mathbf{s}_n, \hat{\mathbf{s}}_1; \psi_j), \ldots, K(\mathbf{s}_n, \hat{\mathbf{s}}_{N_c}; \psi_j)\Big]$, it can be verified that

$$\log\big(1 - \sigma(g_k^j(\mathbf{s}_n))\big) = -\mathbf{W}_k^T \mathbf{x}_n^j + \log\sigma(g_k^j(\mathbf{s}_n)). \tag{5.16}$$

Inserting (5.16) into the kernel width's posterior distribution, (5.15) can be reduced to

$$\begin{aligned} p(\psi_k = \psi_j^* | \cdots) \propto \;& U_j \cdot \exp\big\{ \sum_n < \xi_{nk} > \big[< \log\sigma\big(g_k^j(\mathbf{s}_n)\big) > \big] \big\} \\ & \cdot \exp\big\{ \sum_n \sum_{l>k} < \xi_{nl} > \big[- < \mathbf{W}_k >^T \mathbf{x}_n^j + < \log\sigma\big(g_k^j(\mathbf{s}_n)\big) > \big] \big\}, \end{aligned} \tag{5.17}$$

in which $< \log\sigma\big(g_k^j(\mathbf{s}_n)\big) >$ is calculated via the variational bound of the logistic sigmoid function in (5.13).

Because of the sampling of the kernel width within the VB iterations, the lower bound shown in (5.12) does not monotonically increase in general. Until the model converges, the lower bound generally has small fluctuations, as shown when presenting experimental results.

5.5 Experiment Results

The LSBP model proposed here may be employed in many problems for which one has spatially-dependent data that must be clustered or segmented. Since the spatial relationships are encoded via a kernel distance measure, the model can also be used to segment time-series data. Below we consider three examples: (*i*) a simple "toy" problem that allows us to compare with related approaches in an easily understood setting, (*ii*) segmentation of multiple speakers in an audio signal, and (*iii*) segmentation of images. When presenting (*iii*), we first consider processing single images, to demonstrate the quality of the segmentations, and to provide more details on the model. We then consider joint segmentation of multiple images, with the goal of inferring relationships between images (of interest for image sorting and search). In all examples the Student-t construction is used to impose the model sparseness, and all model coefficients (including the bias terms) are drawn from the same prior.

5.5.1 Setting model parameters

To implement LSBP, one must set several parameters. As discussed above, the hyperparameters associated with the Student-t prior on w_{ki} are set as $a_0 = b_0 = 10^{-6}$, this corresponding to the settings of the related RVM [BT00]. The number of kernel centers N_c is generally set in a natural manner, depending upon the application. For example, in the audio example considered in section 5.5.3, N_c is set to the number of total temporal subsequences used to sample the signal. For the image-processing application, N_c may be set to the number of superpixels used to define space-dependent image features (discussed in more detail when presenting image-segmentation results in section 5.5.4). The truncation level K on the LSBP may be set to any large value that exceeds the number of anticipated segments in the image, or model selection may be employed to select K.

We must also define a set of possible kernel scales, $\{\psi_j^*\}_{j=1}^{\tau}$. These again are set naturally to define the relative range of scales in the data under test. For example, in the image-segmentation application, we select τ scale levels to cover a range of resolutions characteristic of the images of interest (*e.g.*, defined by the size of the expected segment sizes relative to the overall image size). In the specific audio and image segmentation applications discussed below we explicitly define these parameters, and note that no tuning of these parameters was performed. Our experience is that any "reasonable" set of kernel scales yields very similar results.

The final thing that must be set within the model is the base measure H. For the audio-signal example the data observed at each time point is a real vector, and therefore it is convenient to use a multivariate Gaussian distribution to represent $F(\boldsymbol{\theta}_n)$. Therefore, in that example the observation-model parameters correspond to the mean and covariance of a Gaussian, implying that the measure H should be a Gaussian-Wishart prior (or a Gaussian-Gamma prior, if a diagonal covariance matrix is assumed in the prior). For the image processing application the observed image feature vectors are quantized, and consequently the observation at any point in the image corresponds to a code index. In this case $F(\boldsymbol{\theta}_n)$ is represented by a multinomial distribution, and hence H is made to correspond to a Dirichlet distribution. Therefore, one may naturally define H based upon the form of the model $F(\cdot)$, in ways typically employed within such Bayesian models.

5.5.2 Simulation Example

In this example the feature vector $\mathbf{y}_n$ is the intensity value of each pixel, and the pixel location is the spatial information $\mathbf{s}_n$. Each observation is assumed to be drawn from a spatially dependent Gaussian mixture (*i.e.*, $F(\cdot)$ is a Gaussian). A comparison is made between the proposed LSBP, the Dirichlet process (DP), and the kernel stick-breaking process (KSBP); for the KSBP, we use the same model as considered in [AWS$^+$08], and this simple example was also taken from that paper. The data are shown in Figure 5.4(a), in which four distinct contiguous sub-regions reside in a background, with a color bar encoding the pixel amplitudes. Each pixel is drawn from a Gaussian distribution with a standard deviation of 10; the two pairs of contiguous regions are generated respectively from the Gaussian distributions with mean intensities equal to 40 and 60, and the background has a mean of 5 [AWS$^+$08]. In the LSBP, DP, and KSBP analyses, we do not set the number of clusters *a priori* and the models infer the number of clusters automatically from the data. Therefore, we fixed the truncation level to $K = 10$ for all models, and the clustering results are shown in Figure 5.4, with different colors representing the cluster index (mixture component to which a data sample is assigned).

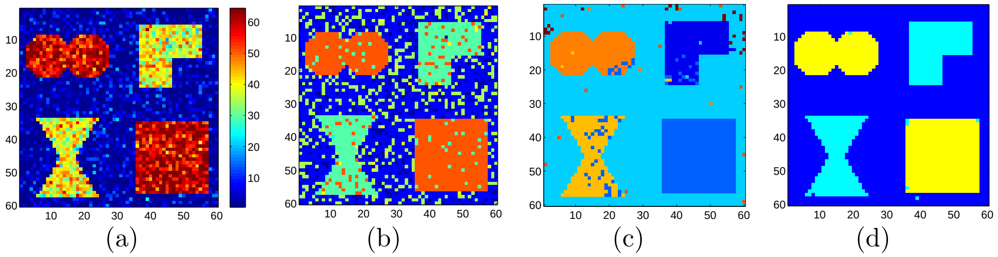
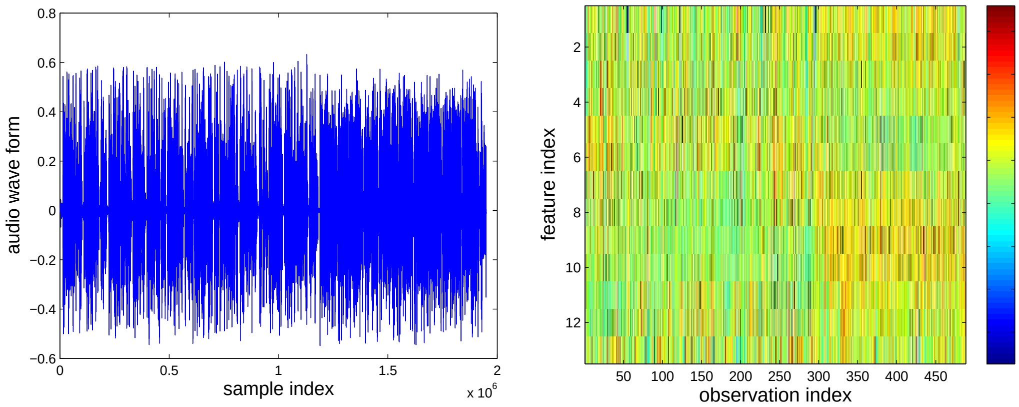


FIGURE 5.4: Segmentation results for the simulation example. (a) original image, (b) DP, (c) KSBP, (d) LSBP

Compared with DP and KSBP, the proposed LSBP shows a much cleaner segmentation in Figure 5.4(d), as a consequence of the imposed favoring of contiguous segments. We also note that the proposed model inferred that there were only three important k (three dominant sticks) within the observed data, consistent with the representation in Figure 5.4(a).



(a) (b)

FIGURE 5.5: Original audio waveform, (a), and representation in terms of MFCC features, (b).

5.5.3 Segmentation of Audio Waveforms

With the kernel in (5.3) specified in a temporal (one-dimensional) space, the proposed LSBP is naturally extended to segmentation of sequential data, such as for speaker diarization [BBBG04,TR06,FSJW08a]. Provided with a spoken document consisting of multiple speakers, speaker diarization is the process of segmenting the audio signal into contiguous temporal regions, and within a given region a particular individual is speaking. Further, one also wishes to group all temporal regions in which a specific individual is speaking.

We assume the acoustic observations at different times are drawn from a Gaussian mixture model (each generating Gaussian ideally corresponds to a speaker ID). Within LSBP and KSBP, the observations of adjacent temporal points are encouraged to be drawn from the same Gaussian, since they are with high probability assumed to be generated from the same source (speaker). The total number of speakers is unknown in advance, and is inferred from the data. An alternative approach, to which we compare, is a sticky HMM [FSJW08a], in which the speech is represented by an HMM with Gaussian state-dependent emissions; to associate a given speaker with a particular state, the states are made to be persistent, or "sticky", with the state-dependent degree of stickiness also inferred.

We consider identification of different speakers from a recording of broadcast news, which may be downloaded with its ground truth[1]. The spoken document has a length of 122.05 seconds, and consists of three speakers. Figure 5.5(a) presents the audio waveform with a sampling rate of 16000 Hz. The ground truth indicates that Speaker 1 talked within the first 13.77 seconds, followed by Speaker 2 until the 59.66 second, then Speaker 1 began to talk again until 74.15 seconds, and Speaker 3 followed and speaks until the end.

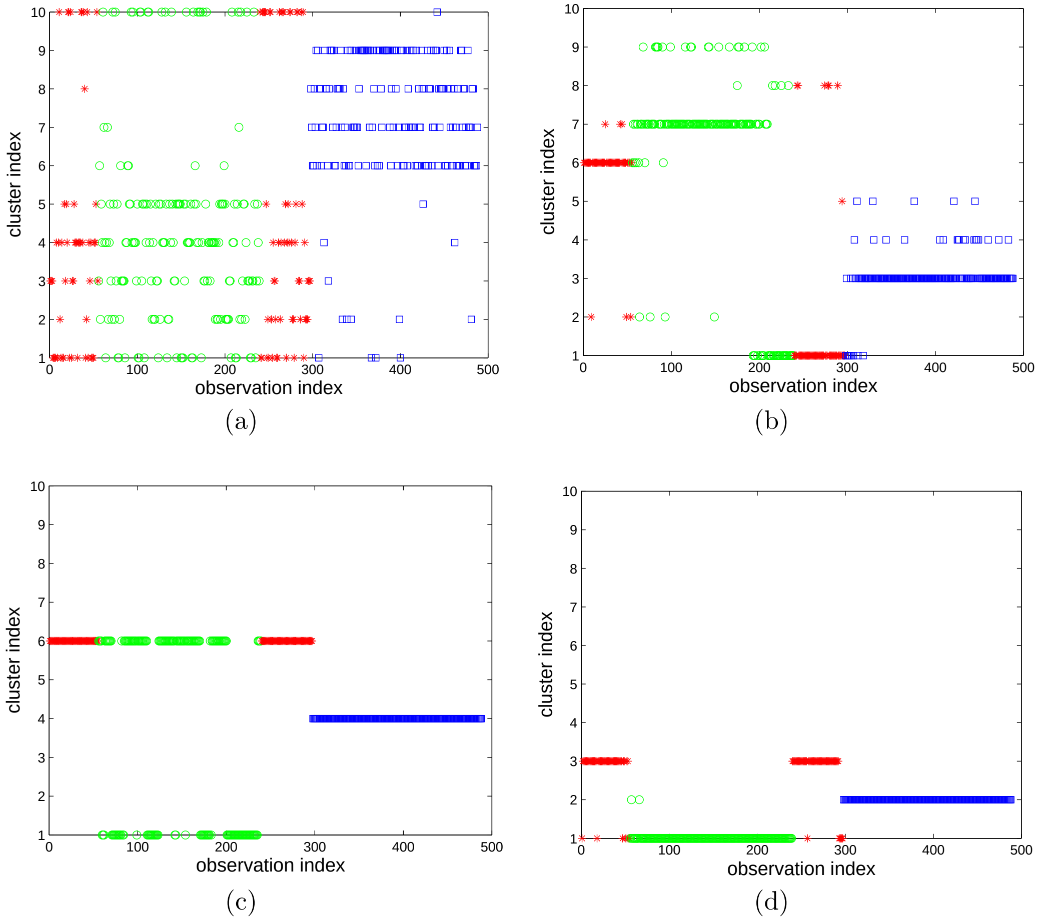
For the feature vector, we computed the first 13 Mel Frequency Cepstral Coefficients (MFCCs) [GFK05] over a 30 ms window every 10 ms, and defined the observations as averages over every 250 ms block, without overlap. We used the first 13 MFCCs because the high frequency content of these features contained little discriminative information [FSJW08a]. The software that we used to extract the MFCCs feature can be downloaded online[2]. There are 488 feature vectors in total, shown in Figure 5.5(b); the features are normalized to zero mean and the standard deviation is made equal to one.

To apply the DP, KSBP and LSBP Gaussian mixture models on this data, we set the truncation level as $K = 10$. To calculate the temporal distance between each pair of observations, we take the observation index from 1 to 488 as the location coordinates in (5.3) for $\mathbf{s}$. The potential kernel-width set is $\mathbf{\Psi}^* = \{50, 100, \ldots, 1000\}$ for LSBP and KSBP; note that these are the same range of parameters used to present the generative model in Figure 5.2. The experiment shows that all the models converge after 20 VB iterations.

For the sticky HMM, we employed two distinct forms of posterior computation:

[1] http://www.itl.nist.gov/iad/mig//tests/rt/2002/index.html

[2] http://www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat/



(a) (b) (c) (d)

FIGURE 5.6: Segmentation results for the audio recording. The colored symbols denote the ground truth: red represents Speaker 1, green represents Speaker 2, blue represents Speaker 3. Each MFCC feature vector is assigned to a cluster index ($K = 10$), with the index shown along the vertical axis. (a) DP, (b) KSBP, (c) sticky HMM using VB inference, (d) LSBP

(i) a VB analysis, which is consistent with the methods employed for the other models; and (ii) a Gibbs sampler, analogous to that employed in the original sticky-HMM paper [FSJW08a]. For both the VB and Gibbs sampler, a truncated stick-breaking representation was used for the DP draws from the hierarchical Dirichlet process (HDP); see [FSJW08a] for a discussion of how the HDP is employed in this model.

To segment the audio data, we labeled each observation to the index of the cluster

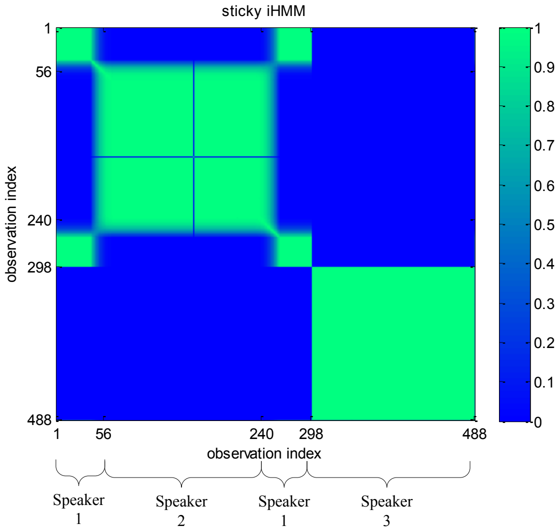


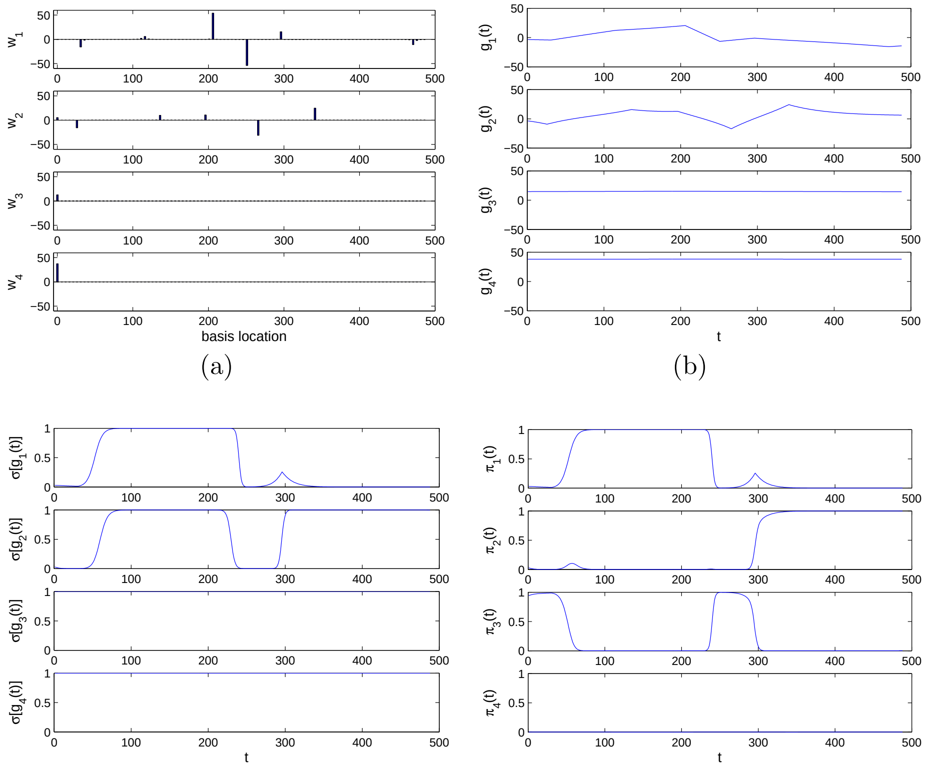
FIGURE 5.7: Sticky HMM results for the data in Figure 5.5(a), based on a Gibbs sampler. The figure denotes the fraction of times within the collection samples that a given portion of the waveform shares the same underlying state.

with the largest probability value, and the results are shown in Figure 5.6 (here the sticky-HMM results were computed via VB analysis). To indicate the ground truth, different symbols and colors are used to represent different speakers.

From the results in Figure 5.6, the proposed LSBP yields the best segmentation performance, with results in close agreement with ground truth. We found the sticky-HMM results to be very sensitive to VB initialization, and the results in Figure 5.6 were the best we could achieve.

While the sticky HMM did not yield reliable VB-computed results, it performed well when a Gibbs sampler was employed (as in [FSJW08a]). In Figure 5.7 are shown the fraction of times within the collection Gibbs samples that a given portion of the signal share the same underlying state; note that the results are in very close agreement with "truth". We cannot plot the Gibbs results in the same form as the VB results in Figure 5.6 due to label switching within the Gibbs sampler. The Gibbs-sampler results were computed using 5000 burn iterations and 5000 collection iterations.

These results demonstrate that the proposed LSBP, based on a fast VB solution, yields results commensurate with a state-of-the-art method (the sticky HMM based on a Gibbs sampler). On the same PC, the VB LSBP results required approximately 45 seconds of CPU time, while the Gibbs sticky-HMM results required 3 hours; in both cases the code was written in non-optimized Matlab, and these numbers should be viewed as providing a *relative* view of computational expense. The accuracy and speed of the VB LSBP is of interest for large-scale problems, like those considered in the next section. Further, the LSBP is a general-purpose algorithm, applicable to time- and spatially-dependent data (images), while the sticky HMM is explicitly designed for time-dependent data.



(a)

(b)

(c)

(d)

FIGURE 5.8: Example draw from the LSBP posterior, for the audio data under test. (a) $\boldsymbol{w}_k$, (b) $g_k(t)$, (c) $\sigma_k(t)$, (d) $\pi_k(t)$

In the LSBP, DP, and KSBP analyses, we do not set the number of clusters *a priori* and the models infer the number of clusters automatically from the data. Therefore, we fixed the truncation level to $K = 10$ for all models, and the clustering results are shown in Figure 5.4, with different colors representing the cluster index (mixture component to which a data sample is assigned).

In Figure 5.2 we illustrated a draw from the LSBP prior, in the absence of any data. The parameters of that example (number of samples, the definition of N_c, and the library $\mathbf{\Psi}^*$) were selected as to correspond to this audio example. To generate the draws in Figure 5.2, a spike-and-slab prior was employed, since the Student-t prior would prefer (in the absence of data) to set all coefficients to zero (or near zero), with high probability. Further, for related reasons we treated the bias terms w_{k0} distinct from the other coefficients. We now consider a draw from the LSBP *posterior*, based on the audio data considered above. This gives further insight into the machinery of the LSBP. We also emphasize that, in this example based on real data, as in all examples shown in this section, we impose sparseness via the Student-t prior. Therefore, when looking at the posterior, we may see which coefficients w_{ki} have been "pulled" away from zero such that the model fits the observed data. A representative draw from the LSBP posterior is shown in Figure 5.8, using the same presentation format as applied to the draw from the prior in Figure 5.2. Note that only three sticks have appreciable probability for any time t, and the segments tend to be localized, with near-unity probability of using a corresponding model parameter within a given segment. While the spike-slab prior was needed to manifest desirable draws from the prior alone, the presence of data simplifies the form of the LSBP prior, based only on a relatively standard use of the hierarchical Student-t construction.

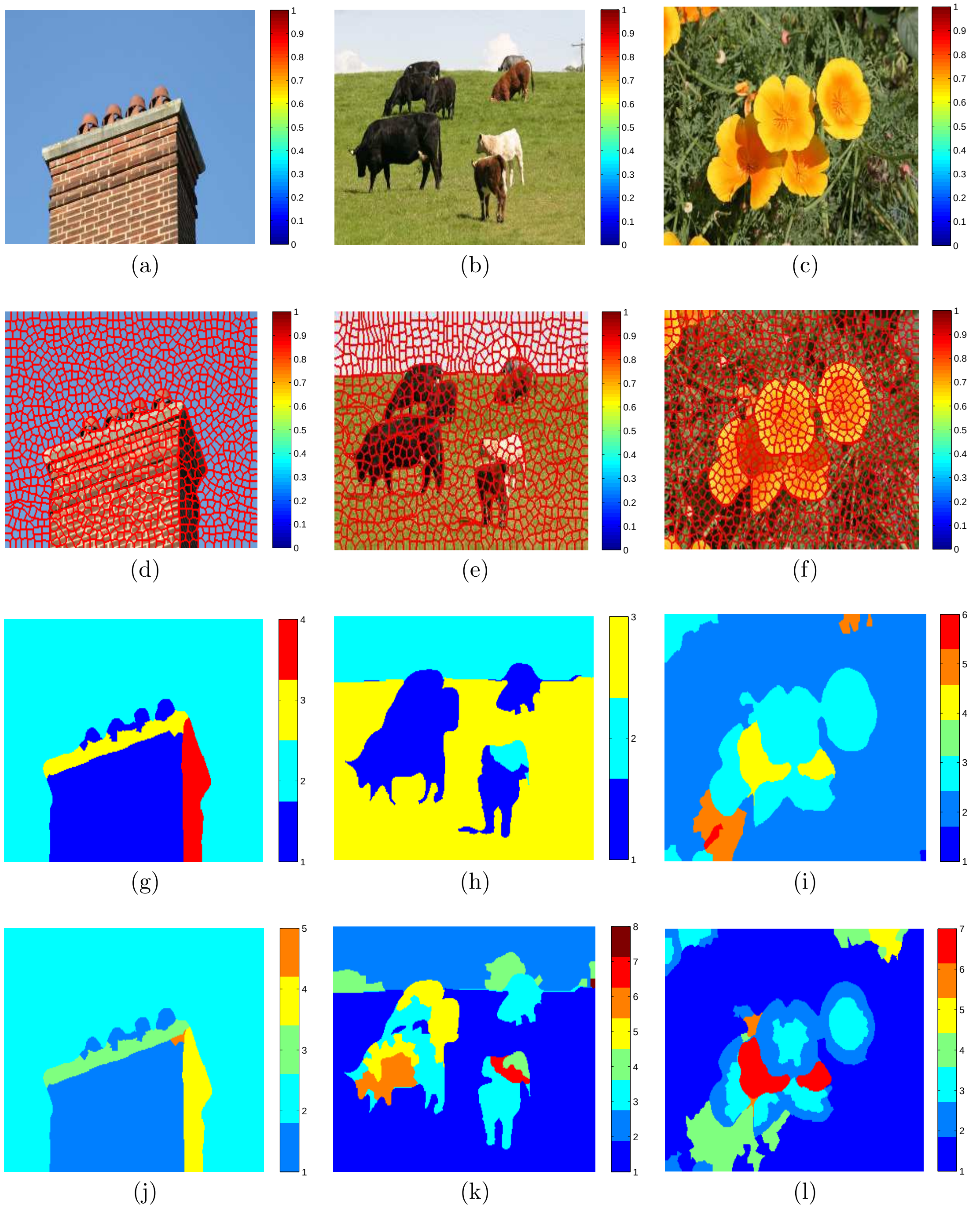
5.5.4 Image Segmentation with LSBP

The images considered first are from Microsoft Research Cambridge[3] and each image has 320×213 pixels. To apply the hierarchical model to image segmentation, we first over-segment each image into $1,000$ "superpixels", which are local, coherent and preserve most of the structure necessary for segmentation at the scale of interest [RM03]. The software used for this is described in [Mor05], and can be downloaded at http://fas.sfu.ca/∼mori/research/superpixels/. Each superpixel is represented by both color and texture descriptors, based on the local RGB, hue feature vectors [WS06], and also the values of Maximum Response (MR) filter banks [VZ02]. We discretize these features using a codebook of size 32, and then calculate the distributions [AP09] for each feature within each superpixel as visual words [CFF07, WG07].

Since each superpixel is represented by three visual words, the mixture components $\boldsymbol{\theta}_k^*$ are three multinomial distributions as $\{\text{Mult}(\mathbf{p}^{1*}_k) \otimes \text{Mult}(\mathbf{p}^{2*}_k) \otimes \text{Mult}(\mathbf{p}^{3*}_k)\}$ for $k = 1, \dots, K$. The variational distribution $q(\boldsymbol{\theta}_k^*)$ is $\text{Dir}(\mathbf{p}^{1*}_k | \tilde{\boldsymbol{\beta}}_k^1) \otimes \text{Dir}(\mathbf{p}^{2*}_k | \tilde{\boldsymbol{\beta}}_k^2) \otimes \text{Dir}(\mathbf{p}^{3*}_k | \tilde{\boldsymbol{\beta}}_k^3)$, and within VB inference we optimize the parameters $\tilde{\boldsymbol{\beta}}_k^1$, $\tilde{\boldsymbol{\beta}}_k^2$, and $\tilde{\boldsymbol{\beta}}_k^3$.

To perform segmentation at the patch level (each superpixel corresponds to one patch), the center of each superpixel is recorded as the location coordinate $\mathbf{s}_n$. The discrete kernel-width set $\boldsymbol{\Psi}^*$ is composed of $30, 35, \dots, 160$, which are scaled empirically based on the image and object average size. Typically we may choose the $\boldsymbol{\Psi}^*$ as a subset between the minimum and maximum Euclidean distance associated with any two data points' spatial locations within this image. To save computational resources, we chose as basis locations $\{\hat{\mathbf{s}}_i\}_{i=1}^{N_c}$ the spatial centers of every tenth superpixel in a given image, after sequentially indexing the superpixels (we found that if we do not perform this subsampling, very similar segmentation results are achieved, but at greater computational expense).

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

FIGURE 5.9: LSBP Segmentation for three image examples. (a)∼(c) image examples of "chimney","cows" and "flowers"; (d)∼(f) image examples represented with "superpixels"; (g)∼(i) "optimal" segmentation results with model selection ($K = 4$ for "chimney", $K = 3$ for "cows" and $K = 6$ for "flowers"); (j)∼(l) segmentation results with a initialization of $K = 10$ for the image examples.

Three representative example images are shown in Figures 5.9(a), (b) and (c); the superpixels are generated by over-segmentation [Mor05] on each image, with associated over-segmentation results shown in Figures 5.9(d), (e) and (f). The segmentation task now reduces to grouping/clustering the superpixels based on the associated image feature vector and associated spatial information. To examine the effect of the truncation level K, and to investigate model selection, we considered K from 2 to 10 and quantified the VB approximation to the model evidence (marginal likelihood). The segmentation performance for each of these images is shown in Figure 5.9(g), (h) and (i), using respectively $K = 4$, 3 and 6, based on the model evidence (discussed further below). These (typical) results are characterized by homogeneous segments with sharp boundaries. In Figure 5.9(j), (k) and (l), the segmentation results are shown with K fixed at $K = 10$. In this case the LSBP has ten sticks, and there is no model selection; however, based on the segmentation there are a subset of sticks (5, 8 and 7, respectively) inferred to have appreciable amplitude.

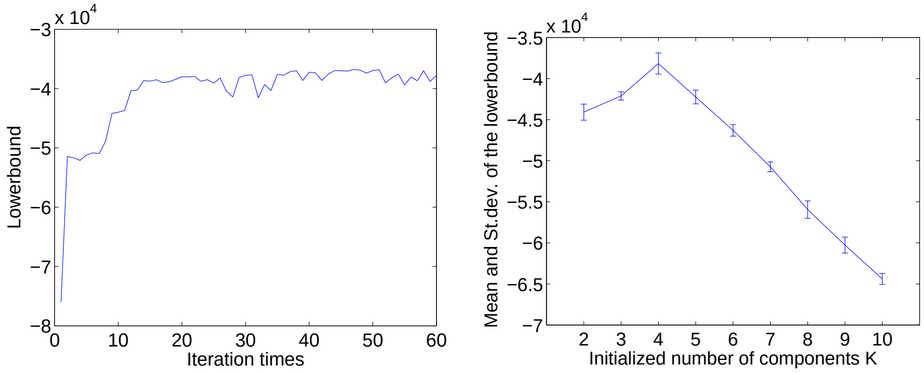
Based upon these representative example results, which are consistent with a large number of tests on related images, we make the following observations. Considering first the “chimney” results in Figure 5.9(a), (g) and (j), for example, we note that there are portions of the brick that have textural differences. However, the prior tends to favor contiguous segments, and one solid texture is manifested for the bricks. We also note the sharp boundaries manifested in the segments, despite the fact that the logistic-regression construction is only using simple Gaussian kernels (not particularly optimized for near-linear boundaries). For the relatively simple “chimney” image, the segmentation results are very similar based on model selection for K (Figure 5.9(g)) and simply truncating the sticks at a “large” value (Figure 5.9(j) with $K = 10$).

The “cow” example is more complex, pointing out further characteristics of LSBP. We again observe homogeneous contiguous segments with sharp boundaries. In this case the model selection yields (as expected) a simpler segmentation (Figure 5.9(h)).

All of the relatively dark cows are segmented together. By contrast, without model selection and with $K = 10$, the results in Figure 5.9(k) capture more details in the cows. However, we also note that in Figure 5.9(k) the clouds are properly assigned to a distinctive type of segment, while in Figure 5.9(h) the clouds are just included in the sky cluster/segment. Similar observations are also obtained from the "flower" example for Figure 5.9(c), with more flower texture details kept with a large truncation level setting in Figure 5.9(l) than the model selection result shown in Figure 5.9(i). It is therefore generally observed that simpler segmentations are manifested by the model-selection procedure.

Because of the sampling of the kernel width, the lower bound of the log model evidence did not increase monotonically in general. For the "chimney" example considered in Figure 5.9(a), the log model evidence was found to sequentially increase approximately within the first 20 iterations and then converge to the local optimal solution with small fluctuations, as shown in Figure 5.10(a) with a model of $K = 4$. To perform model selection, we calculate the mean and standard deviation of the lower bound after 25 iterations for each K, as plotted in Figure 5.10(b); from this figure one clearly observes that the data favor the model with $K = 4$, for at this point the VB lower bound (approximation to the evidence) has its largest value. Hence, one may stop examining increasing K once it is evident that the model evidence is falling with increasing K (as compared with simply setting K to a large value).

To further evaluate the performance of LSBP for image segmentation, we also consider several other state-of-art methods, including two other non-parametric statistical models: the Dirichlet process (DP) [Set94] and the kernel stick-breaking process (KSBP) [AWS+08]. We also consider two graph-based spectral decomposition methods: normalized cuts (Ncuts) [SM00] and multi-scale Ncut with long-range graph connections [CBS05]. Further, we consider the Student-t distribution mixture model (Stu.-t MM) [SNG07], and also spatially varying mixture segmentation with



(a) (b)

FIGURE 5.10: LSBP Segmentation for three image examples. (a)VB iteration lowerbound for image "chimney" with $K = 4$; (b) Model selection as a function of K for image "chimney".

edge preservation (St.-svgm) [SNP08]. We consider the same data source as in the previous examples, but for the next set of results segmentation "ground truth" was provided with the data. The data are divided into eight categories: trees, houses, cows, faces, sheep, flowers, lake and street; each category has thirty images. All models were initialized with a segment number of $K = 10$.

Figure 5.11 shows typical segmentation results for the different algorithms. Given a segment count number, both the normalized cuts and the multi-scale Ncut produced very smooth segmentations, while certain textured regions might be split into several pieces. The Student-t distribution mixture model (Stu.-t MM) yields a relatively robust segmentation, but it is sensitive to the texture appearance. Compared with Stu.-t MM, the spatially varying mixtures (St.-svgm) favors a more contiguous segmentation for the texture region, preserving edges; this may make a good tradeoff between keeping coherence and capturing details, but the segmentation performance is degraded by redundant boundaries, such as those within the goose body. Compared with these state-of-art algorithms, the LSBP results appear to be very competitive. Among the Bayesian methods (DP, KSBP and LSBP), LSBP tends to yield better segmentation, characterized by homogeneous segmentation regions and

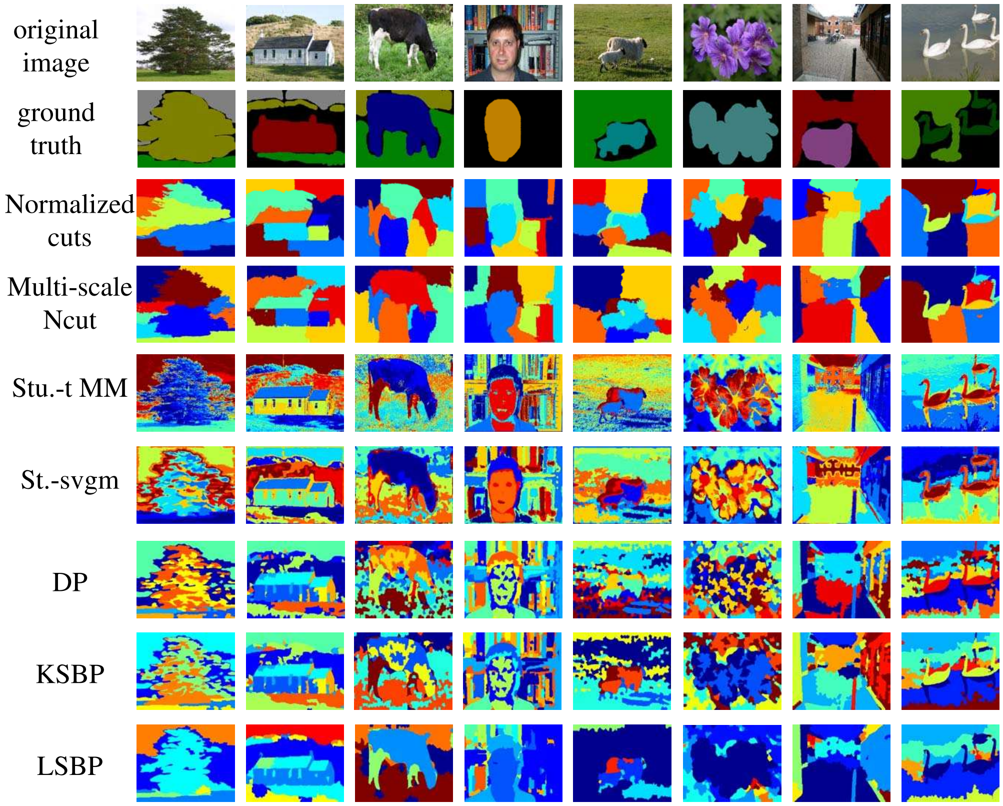


FIGURE 5.11: Segmentation examples of different methods with an initialization of $K = 10$. From top to down, each row shows: the original image, the image ground truth, normalized cuts, multiscale Ncut, Student-t distributions mixture model (Stu.-t MM), spatially varying mixtures (St.-svgm), DP mixture, KSBP mixture, and the LSBP mixture model.

sharp segment boundaries.

To quantify segmentation results, we also calculated the Random Index (RI) [UPH07] and the Variation of Information (VoI) [Mei03], using segmentation "truth" provided with the data. RI measures consistency between two segmentation labels via an overlapping fraction, and VoI roughly calculates the amount of randomness that exists in one segmentation that is not explained by the other. Accordingly, for the RI measure, larger values represent better performance, and for VoI smaller values are preferred. We calculated the average RI and VoI values of the thirty images for

K		2	4	6	8	10
Ncuts	mean	0.5552	0.6169	0.6269	0.6180	0.6093
	median	0.5259	0.6098	0.6376	0.6286	0.6235
	st. dev.	0.0953	0.1145	0.1317	0.1402	0.1461
Multi-scale Ncuts	mean	0.6102	0.6491	0.6387	0.6306	0.6228
	median	0.5903	0.6548	0.6515	0.6465	0.6396
	st. dev.	0.0979	0.1361	0.1462	0.1523	0.1584
Stu.-t MM	mean	0.6522	0.6663	0.6409	0.6244	0.6110
	median	0.6341	0.6858	0.6631	0.6429	0.6360
	st. dev.	0.1253	0.1248	0.1384	0.1455	0.1509
St.-svgm	mean	**0.6881**	**0.6861**	0.6596	0.6393	0.6280
	median	**0.6781**	**0.7026**	0.6825	0.6575	0.6516
	st. dev.	0.1249	0.1262	0.1427	0.1532	0.1599
DP	mean	0.6335	0.6527	0.6389	0.6270	0.6187
	median	0.6067	0.6669	0.6431	0.6321	0.6232
	st. dev.	0.1272	0.1283	0.1384	0.1464	0.1507
KSBP	mean	0.6306	0.6530	0.6396	0.6290	0.6229
	median	0.5963	0.6693	0.6448	0.6371	0.6272
	st. dev.	0.1237	0.1303	0.1397	0.1464	0.1523
LSBP	mean	0.6516	0.6791	**0.6804**	**0.6704**	**0.6777**
	median	0.6384	0.6921	**0.6900**	**0.6835**	**0.6885**
	st. dev.	0.1310	0.1202	0.1263	0.1294	0.1319

Table 5.1: Statistics on the averaged Rand Index (RI) over 240 images as a function of K (Microsoft Research Cambridge images).

each category; the statistics for the two measures are depicted in Tables 5.1 and 5.2, considering all 240 images and various K.

Compared with other state-of-the-art methods, the LSBP yields relatively larger mean and median values for average RI, and relatively small average VoI, for most K. For $K = 2$ and 4 the spatially varying mixtures (St.-svgm) shows the largest RI values, while it does not yield similar effectiveness as K increases. In contrast, the LSBP yields a relatively stable RI and VoI from $K = 4$ to 10. This property is more easily observed in Figure 5.12, which shows the averaged RI and VoI evaluated as a function of K, for categories "houses" and "cows". The Stu.-t MM, St.-svgm, DP and KSBP have similar performances for most K; LSBP generates a competitive

K		2	4	6	8	10
Ncuts	mean	1.7911	2.2034	2.4344	2.6885	2.8828
	median	1.8201	2.1990	2.4392	2.7134	2.8956
	st. dev.	0.4402	0.4213	0.4003	0.3673	0.3615
Multi-scale Ncuts	mean	1.7017	2.0538	2.3535	2.5548	2.7397
	median	1.7322	2.0238	2.3746	2.5912	2.7471
	st. dev.	0.4253	0.4276	0.4030	0.4056	0.4215
Stu.-t MM	mean	1.4903	2.0078	2.4258	2.7421	3.0085
	median	1.5312	2.0283	2.4653	2.7495	3.0341
	st. dev.	0.5161	0.4544	0.4120	0.3941	0.3798
St.-svgm	mean	1.4031	1.8957	2.2667	2.5764	2.7999
	median	1.4000	1.8957	2.2673	2.5919	2.8123
	st. dev.	0.5094	0.4176	0.4113	0.3956	0.4001
DP	mean	1.4810	1.9522	2.2961	2.5808	2.7740
	median	1.5145	1.9522	2.3541	2.6321	2.8432
	st. dev.	0.4952	0.3923	0.4186	0.4164	0.4573
KSBP	mean	1.4806	1.9383	2.3063	2.5888	2.7873
	median	1.4980	1.9811	2.3403	2.6304	2.8338
	st. dev.	0.4811	0.3919	0.4150	0.4128	0.4457
LSBP	mean	**1.4484**	**1.8142**	**1.9811**	**2.1050**	**2.0861**
	median	**1.4631**	**1.8288**	**1.9825**	**2.1528**	**2.1178**
	st. dev.	0.4835	0.4478	0.4979	0.5101	0.5254

Table 5.2: Statistics on the Variation of Information (VoI) over 240 images as a function of K (Microsoft Research Cambridge images).

result with a smaller K, and also yields robust performance with a large K.

We also considered the Berkeley 300 data set[4]. These images have size 481 × 321 pixels, and we also over-segmented each image into 1000 superpixels. Both the RI and VoI measures are calculated on average, with the multiple labels (human labeled) provided with the data. Each individual image typically has roughly ten segments within the ground truth. We calculated the evaluation measures for $K = 5$, 10 and 15. Table 5.3 presents results, demonstrating that all methods produced competitive results for both the RI and VoI measures. By a visual evaluation of the segmentation results (see Figure 5.13), multi-scale Ncut is not as good as the other methods when the segments are of irregular shape and unequal size.

[4] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

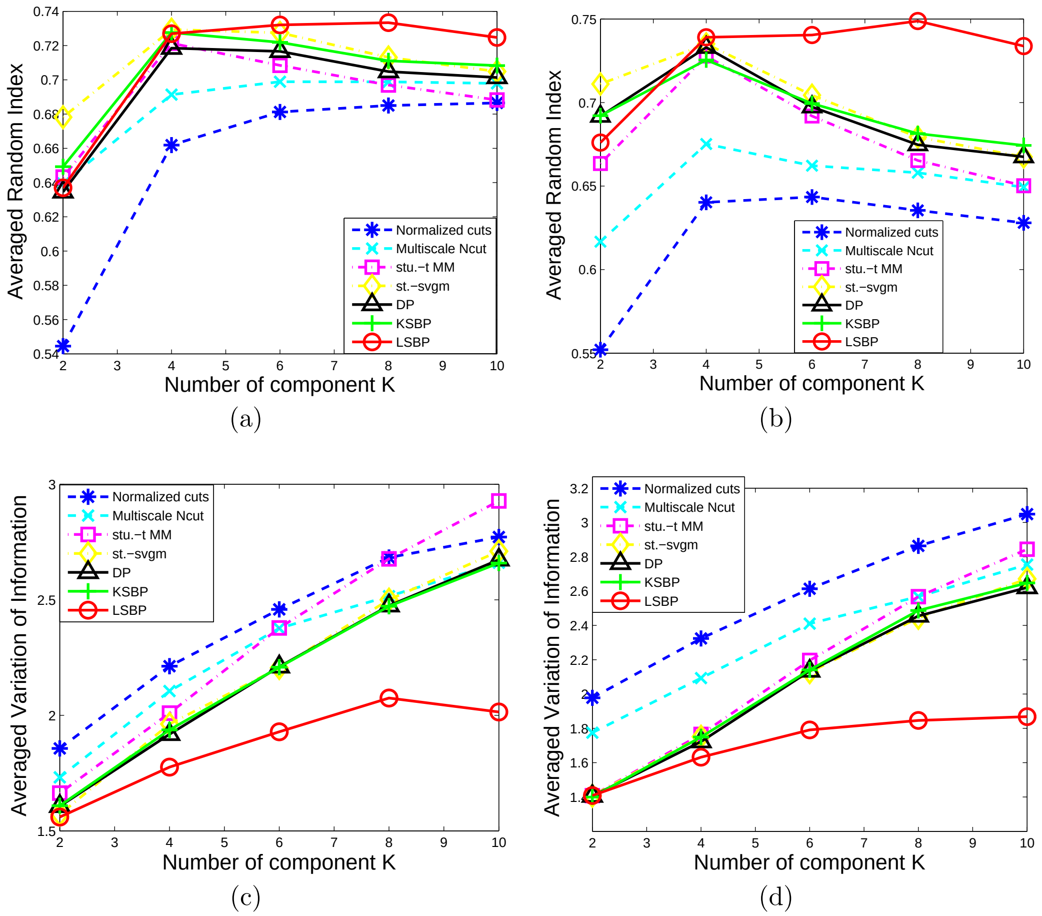


FIGURE 5.12: Average Random Index (RI) and Variation of Information (VoI) as functions of K with image categories. (a) RI for "houses" , (b) RI for "cows", (c) VoI for "houses", (d) VoI for "cows".

	Normalized cuts	Multiscale Ncut	Stu.-t MM	St.-svgm	DP mixture	KSBP mixture	LSBP mixture
RI	0.7220	0.7404	0.7093	0.7188	0.7228	0.7237	0.7241
VoI	2.7857	2.5541	3.7772	3.5682	2.8573	2.7027	2.6591

Table 5.3: Different segmentation methods compared on Berkeley 300 images data set.

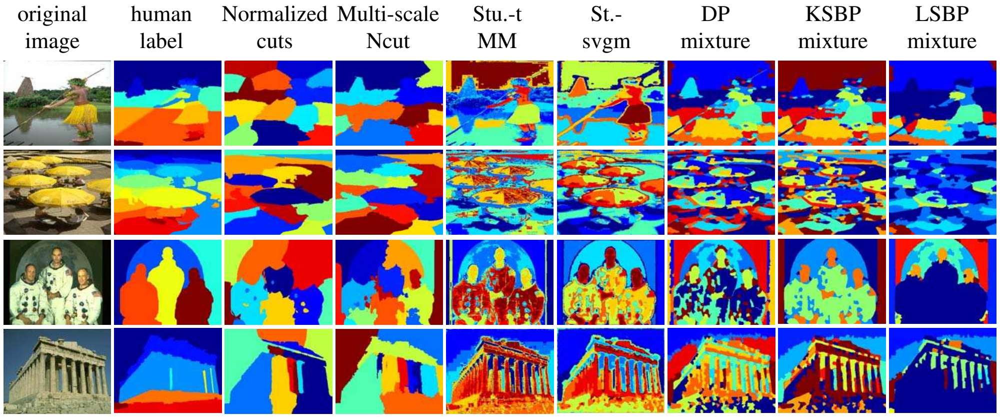


FIGURE 5.13: Segmentation examples of different methods with $K = 10$, for Berkeley image data set. From left to right, each column shows: the original image, the image ground truth, normalized cuts, multiscale Ncut, the Student-t distribution mixture model (Stu.-t MM), spatially varying mixtures (St.-svgm), DP mixture, KSBP mixture, and the LSBP mixture model.

The purpose of this section was to demonstrate that LSBP yields competitive segmentation performance, compared with many state-of-the-art algorithms. It should be emphasized that there is no perfect way of quantifying segmentation performance, especially since the underlying "truth" is itself subjective. An important advantage of the Bayesian methods (DP, KSBP and LSBP) is that they may be readily extended to joint segmentation of multiple images, considered in the next Chapter.

5.6 Summary

The logistic stick-breaking process (LSBP) is proposed for clustering spatially- or temporally-dependent data, imposing the belief that proximate data are more likely to be clustered together. The sticks in the LSBP are realized via multiple kernel-based logistic regression functions, with a shrinkage prior employed for favoring contiguous and spatially localized partitions. Competitive segmentation performance has been manifested in several examples. Relative to other related approaches, the proposed LSBP yields sharp segmentations, and is able to automatically infer an appropriate number of segments.

6

Hierarchical LSBP (H-LSBP)

6.1 Extend the LSBP to Multi-task Learning

In addition to exploiting spatial information when performing clustering, there has also been recent research on the simultaneous analysis of *multiple* tasks. This is motivated by the idea that multiple related tasks are likely to share the same or similar attributes [Car97, AWS$^+$08, PSH08]. Exploiting the information contained in multiple data sets ("tasks"), model-parameter estimation may be improved [TJBB06,PSH08,SJ08]. Therefore, it is desirable to employ multi-task learning when processing multiple spatially-dependent data (*e.g.*, images).

Motivated by previous multi-task research [TJBB06, AWS$^+$08], we consider the problem of simultaneously processing multiple spatially-dependent data sets. A separate LSBP prior is employed for each of the tasks, and all LSBPs share the same base measure, which is drawn from a DP. Hence, a "library" of model parameters – atoms – is shared across all tasks. This construction is related to the hierarchical Dirichlet process (HDP) [TJBB06], and is referred to here as a hierarchical logistic stick-breaking process (H-LSBP).

We employ the H-LSBP to simultaneously segment multiple images. In addition to inferring a segmentation of each image, the framework allows sorting and searching among the images.

6.2 Statistical Model

Multi-task learning (MTL) is an inductive transfer framework [Car97], with the goal of improving modeling performance by exploiting related information in multiple data sets. We here employ MTL for joint analysis of multiple spatially dependent data sets, yielding a hierarchical logistic stick-breaking process (H-LSBP). This framework models each individual data set (task) with its own LSBP draw, while sharing the same set of model parameters (atoms) across all tasks, in a manner analogous to HDP [TJBB06]. The set of shared model atoms are inferred in the analysis.

The spatially-dependent probability measure for task m, G_m, is drawn from a LSBP with base measure G_0, and G_0 is shared across all M tasks. Further, G_0 is drawn from a Dirichlet process [BM73], and in this manner each task-dependent LSBP shares the same set of discrete atoms. The H-LSBP model is represented as

$$\begin{gathered} \mathbf{y}_{mn}|\boldsymbol{\theta}_{mn} \sim F(\boldsymbol{\theta}_{mn}), \quad \boldsymbol{\theta}_{mn}|G_m \sim G_m \\ G_m|\{G_0, a_0, b_0, \boldsymbol{\Psi}^*\} \sim \text{LSBP}(G_0, a_0, b_0, \boldsymbol{\Psi}^*), \\ G_0|\gamma, H \sim \text{DP}(\gamma H), \end{gathered} \tag{6.1}$$

Note that we are assuming a Student-t construction of the sparseness prior within the LSBP, defined by hyperparameters a_0 and b_0.

Assume task $m \in \{1, \ldots, M\}$ has N_m observations, defining the data $\mathbf{D}_m = \{\mathbf{D}_{m1}, \ldots, \mathbf{D}_{m(N_m)}\}$. We introduce a set of latent indicator variables $\{\mathbf{t}_m = t_{m1}, \ldots, t_{m\infty}\}$ for each task, with

$$t_{mk} \overset{iid}{\sim} \sum_{l=1}^{\infty} \beta_l \delta_l, \quad k = 1, \ldots, \infty, \quad m = 1, \ldots, M, \tag{6.2}$$

where β_l corresponds to the lth stick weight of the stick-breaking construction of the DP draw $G_0 = \sum_{l=1}^{\infty} \beta_l \delta_{\boldsymbol{\theta}_l^*}$. The indicator variables t_{mk} establish an association between the observations from each task and the atoms $\{\boldsymbol{\theta}_l^*\}_{l=1}^{\infty}$ shared globally; hence the atom $\boldsymbol{\theta}_{t_{mk}}^*$ is associated with LSBP g_k for task m. Accordingly, we may write the probability measure G_m, for position $\mathbf{s}_{mn}$, in the form

$$G_{\mathbf{s}_{mn}} = \sum_{k=1}^{\infty} \boldsymbol{\pi}_{mk}(\mathbf{s}_{mn}) \delta_{\boldsymbol{\theta}_{t_{mk}}^*}. \tag{6.3}$$

Note that it is possible that in such a draw we may have the same atom used for two different LSBP g_k. This doesn't pose a problem in practice, as the same type of segment (atom) may reside in multiple distinct spatial positions (*e.g.*, of an image), and the different k with the same atom may account for these different regions of the data.

A graphical representation of the proposed hierarchical model is depicted in Figure 6.1. As in the single-task LSBP discussed in Chapter 5, a uniform prior is placed on the discrete elements of $\boldsymbol{\Psi}^*$, and the precision parameter γ for the Dirichlet process is assumed drawn from a gamma distribution $\text{Ga}(e_0, f_0)$. In practice we truncate the number of sticks used to represent G_0, employing $L-1$ draws from the beta distribution, and the length of the Lth stick is $\beta_L = 1 - \sum_{l=1}^{L-1} \beta_l$ [IJ01]. We also set a truncation level K for each G_m, analogous to truncation of a traditional stick-breaking process.

We note that one may suggest drawing L atoms $\boldsymbol{\theta}_l^* \sim H$, for $l = 1, \ldots, L$, and then simply assigning each of these atoms in the same way to each of $\{g_k\}_{k=1}^{K}$ in the M LSBPs associated with the M images under test. Although there are K functions g_k in the LSBP, as a consequence of the stick-breaking construction, those with small index k are more probable to be used in the generative process. Therefore, the process reflected by (6.2) serves to re-order the atoms in an task-dependent manner, such that the important atoms for a given task occur with small index k. In our experiments, we make $K < L$, since the number of different segments/atoms anticipated for any given task is expected to be small relative to the library of possible atoms $\{\boldsymbol{\theta}_l^*\}_{l=1}^L$ available across all tasks.

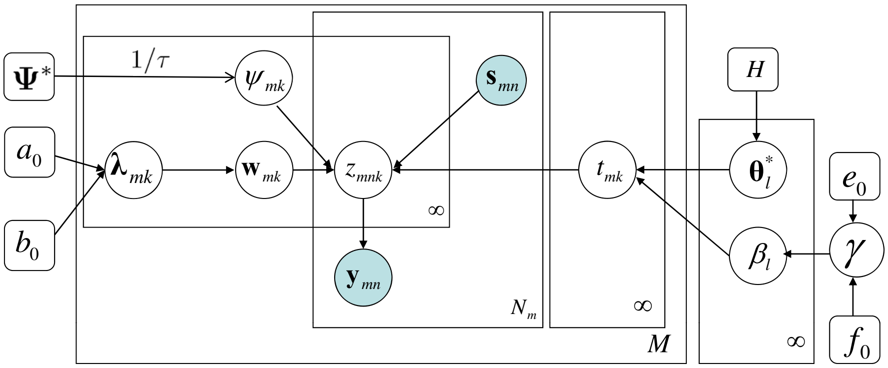


FIGURE 6.1: Graphical representation of H-LSBP.

One may view the H-LSBP model as a hierarchy of multiple layers, in terms of a hierarchical tree structure as depicted in Figure 6.2. In this figure $G_{m1}, \ldots, G_{m(K-1)}$ represent the $K-1$ "gating nodes" within the mth task, and each gating node controls how the data are assigned to the K layers. Thus, the H-LSBP may be viewed as a mixture-of-experts model [BS03] with spatially dependent gating nodes. Given the assigned layer k indicated by $\mathbf{z}_{mn}$, the appearance feature $\mathbf{y}_{mn}$ is drawn from the associated atom $\boldsymbol{\theta}_{t_{mk}}^*$.

For the H-LSBP results one must also set L, which defines the total library size of model atoms/parameters shared across the multiple data sets. Again, we have found any relatively large setting for L to yield good results, as the nonparametric nature of LSBP manifests a selection of which subset of the L library elements are actually needed for the data under test.

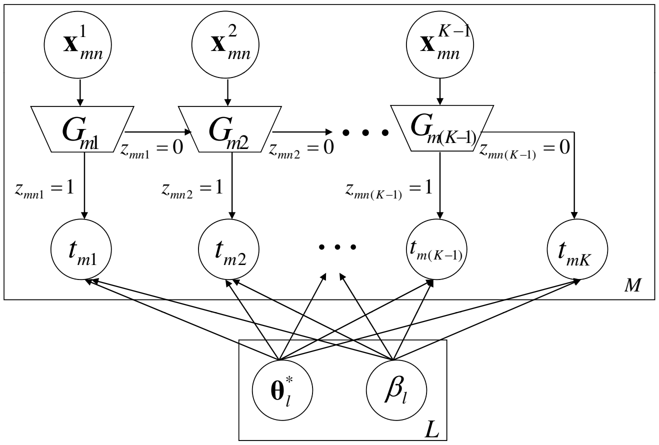


FIGURE 6.2: Hierarchical tree structure representation of the H-LSBP, with spatially dependent gating nodes. The parameters $\mathbf{x}_{mn}^{k}$ are defined as $\mathbf{x}_{mn}^{k} = \{1, \{K(\mathbf{s}_{mn}, \hat{\mathbf{s}}_{mi}; \psi_{mk})\}_{i=1}^{N_c}\}$.

6.3 Model Inference

For the model introduced in section 6.2, we assume

$$q(\mathbf{\Phi}) = q(\gamma)\prod_{l=1}^{L} q(\boldsymbol{\theta}_l)\prod_{l'=1}^{L-1} q(\tilde{\beta}_{l'})\prod_{m=1}^{M}\Big[\prod_{k'=1}^{K} q(t_{mk'})\prod_{k=1}^{K-1}\big[q(\mathbf{w}_{mk})q(\boldsymbol{\lambda}_{mk})\prod_{n=1}^{N_m} q(z_{mnk})\big]\Big], \tag{6.4}$$

where $q(\boldsymbol{\theta}_l)$ is the Dirichlet distribution, the same form as its prior $p(\boldsymbol{\theta}_l|\boldsymbol{\alpha}_0)$. Then $q(\boldsymbol{\theta}_l|\tilde{\boldsymbol{\alpha}}_l)$ is updated with a uniform prior specified for $\boldsymbol{\alpha}_0$ as follows:

$$\tilde{\alpha}_{li} = \alpha_{0i} + \sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k'=1}^{K} < \xi_{mn,k'} >_{q(\mathbf{z}_{mn})} < t_{mk',l} > y_{mni} \tag{6.5}$$

where $\alpha_{0i} = 1/I$ for $i = 1, \ldots, I$, and I is the feature dimension; $< \xi_{mn,k'} >_{q(\mathbf{z}_{mn})} = \prod_{k=1}^{k'-1}(1 - q(z_{mn} = k)) \cdot q(z_{mn} = k')$ represents the approximated posterior probability that data D_{mn} is associated with the hidden "atom" $t_{mk'}$. For $k' = K$, $\xi_{mn,k'} = \prod_{k=1}^{k'-1}(1 - q(z_{mn} = k))$. Finally, $< t_{mk',l} >= q(t_{mk'} = l)$ represents the approximated posterior probability that $t_{mk'}$ takes the atom $\boldsymbol{\theta}_l$.

For updating $q(\tilde{\boldsymbol{\beta}})$ and $q(\gamma)$ given the prior $p(\gamma) = Ga(\gamma|e_0, f_0)$, assume $q(\tilde{\beta}_l) = Be(\tilde{\beta}_l|\pi_{l1}, \pi_{l2})$ with $l = 1, \ldots, L$, and $q(\gamma) = Ga(\gamma|\tilde{e}, \tilde{f})$. Then the update equations are as follows:

$$\begin{array}{l} \pi_{l1} = 1 + \sum_{m=1}^{M} \sum_{k'=1}^{K} < t_{mk',l} > \\ \pi_{l2} = \tilde{e}/\tilde{f} + \sum_{m=1}^{M} \sum_{k'=1}^{K} \sum_{l'=l+1}^{L} < t_{mk',l'} > \\ \tilde{e} = e_0 + L - 1 \\ \tilde{f} = f_0 - \sum_{l=1}^{L-1} \big[\psi(\pi_{l2}) - \psi(\pi_{l1} + \pi_{l2})\big], \end{array} \tag{6.6}$$

in which $\psi(\cdot)$ is the Digamma function.

Given the approximate distribution of the other variables,

$$q(t_{mk'} = l) \propto \exp\Big[< \log p(t_{mk'}|\boldsymbol{\beta}) >_{q(\boldsymbol{\beta})} + < \log p(\mathbf{y}_m|t_{mk'}, \mathbf{z}_m, \boldsymbol{\theta}_l) >_{q(\mathbf{z}_m), q(\boldsymbol{\theta}_l)} \Big], \tag{6.7}$$

where $< \cdot >_{q(\cdot)}$ represents the expectation of the associated variable's distribution. One may readily derive that

$$\begin{aligned} q(t_{mk'} = l) \;\; \propto \;\; & \exp\Big[\sum_{l'=1}^{l-1} \big[\psi(\pi_{l'2}) - \psi(\pi_{l'1} + \pi_{l'2})\big] + \big[\psi(\pi_{l1}) - \psi(\pi_{l1} + \pi_{l2})\big] + \\ & \sum_{n=1}^{N_m} < \xi_{mn,k'} >_{q(\mathbf{z}_{mn})} < \log p(y|\boldsymbol{\theta}_l) >_{q(\boldsymbol{\theta}_l)} \Big], \end{aligned} \tag{6.8}$$

where $< \log p(y|\boldsymbol{\theta}_l) >_{q(\boldsymbol{\theta}_l)}$ is the data likelihood, with expectation performed with respect to the distribution of atoms $\boldsymbol{\theta}_l$ (which may be derived readily). Then $q(t_{mk'}) = \text{Mult}(u_{mk'1}, \ldots, u_{mk'L})$, in which $u_{mk'l} = \frac{q(t_{mk'}=l)}{\sum_{l'=1}^{L} q(t_{mk'}=l')}$.

Similarly, assume $q(\mathbf{W}_{mk}) = N(\tilde{\mathbf{m}}_{mk}, \tilde{\mathbf{\Gamma}}_{mk})$ and $q(z_{mnk} = 1) = \rho_{mn,k} = \sigma(h_{mnk})$ for $k = 1, \ldots, K-1$, then

$$h_{mnk} = \sum_{k'=k}^{K} (-1)^{\nu_{kk'}} < \xi_{mn,k'}^{-k} >_{q(\mathbf{z}_{mn}^{-k})} \sum_{l'=1}^{L} q(t_{mk'} = l')\big[< \log p(y|\boldsymbol{\theta}_l) >_{q(\boldsymbol{\theta}_l)} \big] + \tilde{\mathbf{m}}_{mk}^T \mathbf{x}_{mn}^k, \tag{6.9}$$

where $< \xi^{-k}_{mn,k'} >_{\mathbf{z}^{-k}_{mn}} = \prod_{j=1,j\neq k}^{K-1} \left[\rho_{mn,j}(-1)^{\nu_{jk'}} + \nu_{jk'}\right]$ is the expectation associated the gating variables $\{\mathbf{z}_{mn1}, \ldots, \mathbf{z}_{mn(k-1)}, \mathbf{z}_{mn(k+1)}, \ldots, \mathbf{z}_{mn(K-1)}\}$ except $\mathbf{z}_{mnk}$, with the following definition for $\nu_{kk'}$:

$$\nu_{kk'} = \begin{cases} 0 & \text{if } t_{mk'} \text{ is in the left subtree of } G_{mk} \text{ (see Fig. 6.2)}, \\ 1 & \text{otherwise.} \end{cases} \tag{6.10}$$

Assuming $q(\lambda_{mki}) = Ga(\tilde{a}_{mki}, \tilde{b}_{mki})$, with $i = 0, 1, \ldots, N_c$, the update equations for $q(\mathbf{W}_{mk})$ are as follows:

$$\begin{aligned} \tilde{\mathbf{\Gamma}}_{mk} &= \left[2\textstyle\sum_{n=1}^{N_m} f(\eta_{mnk}) \mathbf{x}^k_{mn} {\mathbf{x}^k_{mn}}^T + \mathrm{diag}(\frac{\tilde{\mathbf{a}}_{mk}}{\tilde{\mathbf{b}}_{mk}})\right]^{-1} \\ \tilde{\mathbf{m}}_{mk} &= \tilde{\mathbf{\Gamma}}_{mk} \textstyle\sum_{n=1}^{N_m} \left[(\rho_{mn,k} - 1/2)\mathbf{x}^k_{mn}\right], \end{aligned} \tag{6.11}$$

where the variational parameter

$$\eta_{mnk} = \sqrt{{\mathbf{x}^k_{mn}}^T (\tilde{m}_{mk}\tilde{m}^T_{mk} + \tilde{\mathbf{\Gamma}}_{mk})\mathbf{x}^k_{mn}}, \tag{6.12}$$

and $f(\eta_{mnk}) = \frac{\tanh(\eta_{mnk}/2)}{4\eta_{mnk}}$ [BT00, BS03]. The parameters $\mathbf{x}^k_{mn}$ are defined as $\mathbf{x}^k_{mn} = \{1, \{K(\mathbf{s}_{mn}, \hat{\mathbf{s}}_{mi}; \psi_{mk}\}_{i=1}^{N_c}\}$.

Given $q(\mathbf{W}_{mk})$, the update equations for $q(\boldsymbol{\lambda}_{mk})$ are

$$\begin{aligned} \tilde{a}_{mki} &= a_0 + 1/2 \\ \tilde{b}_{mki} &= \tfrac{1}{2}(\tilde{\mathbf{\Gamma}}_{mk}(i,i) + \tilde{m}^2_{mki}) + b_0 \end{aligned} \tag{6.13}$$

6.4 Joint Image Segmentation with H-LSBP

In this section we consider H-LSBP for joint segmentation of multiple images. Experiments are performed on the Microsoft data, with another two unlabeled categories: “cloud” and “office”. Each category is composed of 30 images, and therefore there are 300 images in total, analyzed simultaneously. The same feature and image processing techniques are employed as above.

The H-LSBP automatically generates a set of indicator variables $\mathbf{z}_{mn}$ for each superpixel. The probability that the nth superpixel within image m is associated with the kth hidden indicator variable t_{mk}, is represented as $p_k(\mathbf{s}_{mn}) = \sigma(g_k(\mathbf{s}_{mn})) \prod_{l<k}(1-\sigma(g_l(\mathbf{s}_{mn})))$. By integrating out the distribution for each hidden indicator variable t_{mk} drawn from the global set of atoms $\boldsymbol{\theta}_k^*$, we approximate the membership for each superpixel by assigning it to the cluster with largest probability. This "hard" segmentation decision is employed to provide labels for each data point (the Bayesian analysis yields a "soft" segmentation in terms of a full posterior distribution), as employed above when considering one image at a time.

Our goal is to segment all the images simultaneously, sharing model parameters (atoms) across all images. The results of this analysis are used to infer the inter-relationship between different images, of interest for image sorting and search. We set truncation levels $L = 40$ and $K = 10$ (similar results were found for larger truncations, and these parameters have not been optimized). As demonstrated below, the model automatically infers the total number of principal atoms shared across all images, and the number of atoms that dominate the segmentation of each individual image. The learning of these principal atoms, across the multiple images, is an important aspect of the model, so that the associated mixture weights with these atoms for each image can be regarded as a measurable quantity of inter-relationship between images [BNJL03, AWS$^+$08]. Specifically, similar images should have similar distributions over the model atoms. With the same inter-relationship measure generated from the HDP [TJBB06], H-KSBP [AWS$^+$08] and the proposed H-LSBP, we may compare model utility as an image sorting or organizing engine.

To depict how the atoms are shared across multiple images with H-LSBP, we display an atom-usage count matrix in Figure 6.3, in which the size of each square size is proportional to the relative counts of that atom in a given image. Similar atom usage was revealed for HDP and H-KSBP (omitted for brevity), but the H-LSBP generally was more parsimonious in its use of atoms. This is attributed to the fact that the spatial continuity constraint within LSBP encourages a parsimonious representation (a relatively small number of contiguous clusters).

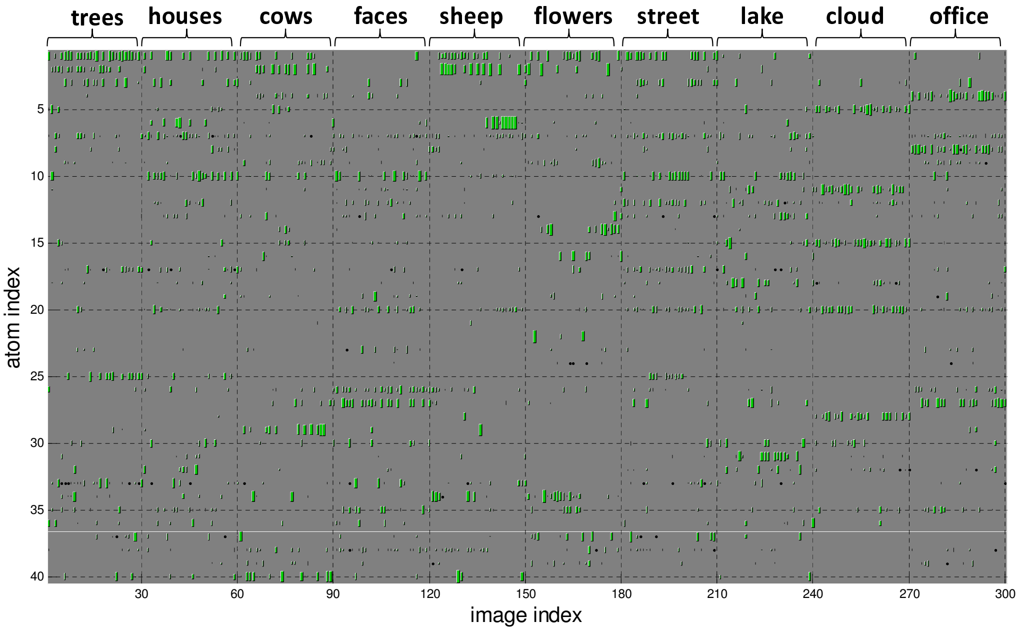


FIGURE 6.3: Atom usage-count matrix for H-LSBP.

Each inferred image atom is in principle associated with one class of features within the images. To get a feel for how the model operates, we examine the types of image segments associated with representative atoms. Specifically, in Figure 6.4 we consider how eight representative atoms are distributed within example images. In this figure we show the original image, and also the same image with all portions *not* associated with a given atom blacked out. From Figure 6.4 we observe that atom 1 is principally associated with trees, atom 2 is associated with grass, atom 4 principally models offices, and atom 10 is mainly attributed to the surface of buildings. Figure 6.5 shows atom examples inferred from the H-KSBP and HDP, and the representative "cloud", "grass", "tree" and "street" atoms do not do as well in maintaining spatial contiguity. This property is especially important to locate certain objects or scenes. For example, for an image annotation task, it is usually expensive to acquire training data set by manually annotating image by image. Therefore, the H-LSBP might

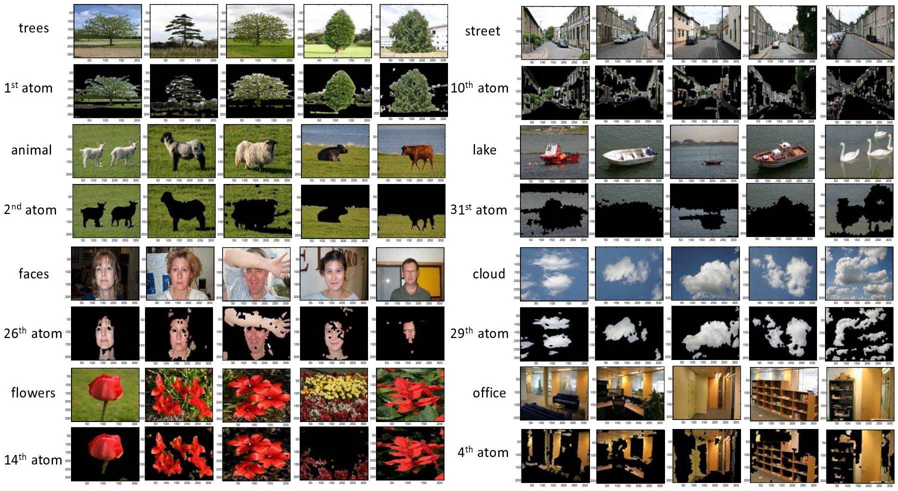


FIGURE 6.4: Demonstration of different atoms inferred by the H-LSBP model. The original images and associated connection to model-parameter atoms are shown on consecutive rows. All regions *not* associated with a respective atom are blacked out.

be used as an automatic annotation tool to save redundant manual work for the preprocessing the images with no words given.

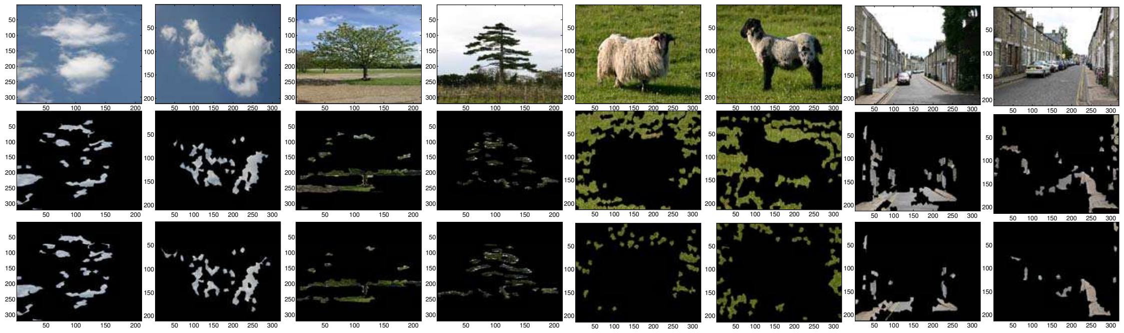


FIGURE 6.5: Examples of different atoms inferred by the H-KSBP and HDP model: The first row is the original images; the second row is the atoms inferred by H-KSBP; the third row is the atoms inferred by HDP.

Based on the atoms inferred from Figure 6.3, we can jointly segment the 300 images with H-LSBP. Each atom represents a label, and the superpixels that shared the same atom are grouped together. Some representative segmentation examples are shown in Figure 6.6, in which each column shows one segmentation example with its "ground truth" (the second row), and the color bar encodes the labels/indexes of the results in the third row (the labels are re-ordered to be different from the atom index).

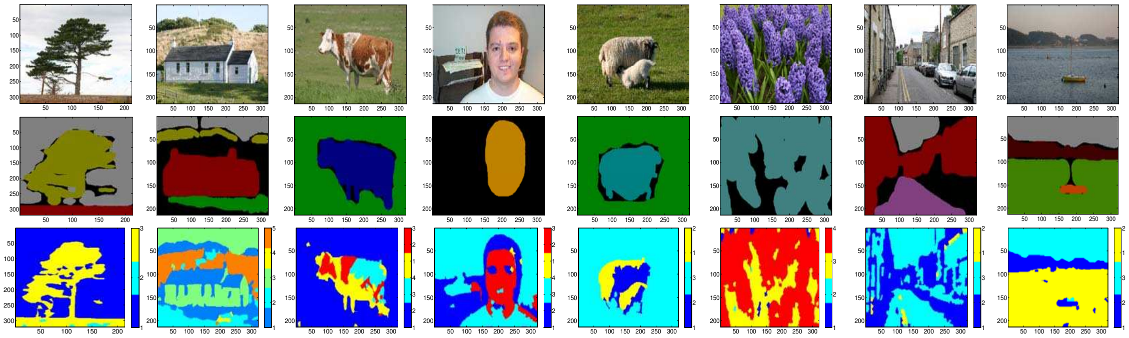
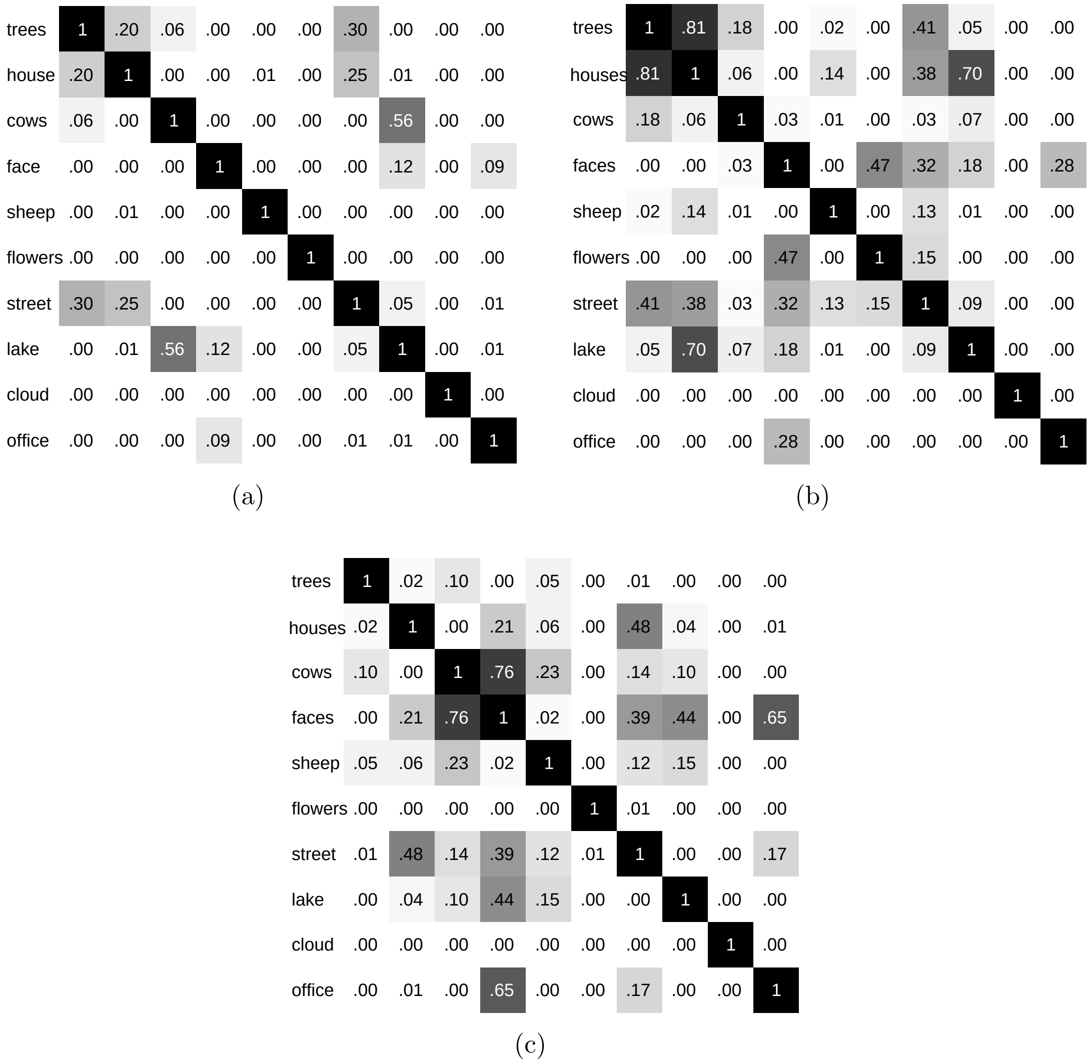


FIGURE 6.6: Representative set of segmentation results of H-LSBP. The top row gives example images, the second row defines "truth" as given by the data set, and the third row represents the respective H-LSBP results.

Another interesting problem is to infer the inter-relationship between different images, and this may be achieved by quantifying the degree to which they share atoms (the sharing of the same set of atoms across all images plays an important role in inferring inter-image relationships). Since we know which atoms $\{\boldsymbol{\theta}_l^*\}_{l=1}^L$ the superpixels within each image are drawn from, we may calculate the Kullback-Leibler (KL) divergence based on the histogram over atoms between each pair of images (a small value is added to the probability of each atom, to avoid numerical problems when computing the KL divergence, when the actual usage of particular atoms may be zero). The KL divergence between different categories, computed by averaging across all of the sub-class images, are shown in Figure 6.7. To make the figure easier to read, the KL divergence D_{KL} is re-scaled as $\exp(-D_{KL})$. In Figure 6.7(a) results are shown based on the proposed H-LSBP, in (b) based upon an H-KSBP analysis, and in (c) based upon an HDP analysis. The H-LSBP, H-KSBP and HDP each yield good results, but Figure 6.7 indicates that the H-LSBP produces smaller cross-class similarity (additionally, the H-KSBP results are better than those of HDP).



trees	1	.20	.06	.00	.00	.00	.30	.00	.00	.00
house	.20	1	.00	.00	.01	.00	.25	.01	.00	.00
cows	.06	.00	1	.00	.00	.00	.00	.56	.00	.00
face	.00	.00	.00	1	.00	.00	.00	.12	.00	.09
sheep	.00	.01	.00	.00	1	.00	.00	.00	.00	.00
flowers	.00	.00	.00	.00	.00	1	.00	.00	.00	.00
street	.30	.25	.00	.00	.00	.00	1	.05	.00	.01
lake	.00	.01	.56	.12	.00	.00	.05	1	.00	.01
cloud	.00	.00	.00	.00	.00	.00	.00	.00	1	.00
office	.00	.00	.00	.09	.00	.00	.01	.01	.00	1

(a)

trees	1	.81	.18	.00	.02	.00	.41	.05	.00	.00
houses	.81	1	.06	.00	.14	.00	.38	.70	.00	.00
cows	.18	.06	1	.03	.01	.00	.03	.07	.00	.00
faces	.00	.00	.03	1	.00	.47	.32	.18	.00	.28
sheep	.02	.14	.01	.00	1	.00	.13	.01	.00	.00
flowers	.00	.00	.00	.47	.00	1	.15	.00	.00	.00
street	.41	.38	.03	.32	.13	.15	1	.09	.00	.00
lake	.05	.70	.07	.18	.01	.00	.09	1	.00	.00
cloud	.00	.00	.00	.00	.00	.00	.00	.00	1	.00
office	.00	.00	.00	.28	.00	.00	.00	.00	.00	1

(b)

trees	1	.02	.10	.00	.05	.00	.01	.00	.00	.00
houses	.02	1	.00	.21	.06	.00	.48	.04	.00	.01
cows	.10	.00	1	.76	.23	.00	.14	.10	.00	.00
faces	.00	.21	.76	1	.02	.00	.39	.44	.00	.65
sheep	.05	.06	.23	.02	1	.00	.12	.15	.00	.00
flowers	.00	.00	.00	.00	.00	1	.01	.00	.00	.00
street	.01	.48	.14	.39	.12	.01	1	.00	.00	.17
lake	.00	.04	.10	.44	.15	.00	.00	1	.00	.00
cloud	.00	.00	.00	.00	.00	.00	.00	.00	1	.00
office	.00	.01	.00	.65	.00	.00	.17	.00	.00	1

(c)

FIGURE 6.7: Similarity matrix associated with the ten image categories. (a) H-LSBP, (b) H-KSBP, (c) HDP

To demonstrate the utility of the proposed method in the context of an image sorting/search engine, we show image sorting examples in Figure 6.8. The left-most column is the original image, and columns 2-6 are the ordered five most similar images in the database, ordered according to the value of the KL divergence between the original image and the remaining 299 images. The five most similar images are shown in Figure 6.8, with generally good sorting performance manifested.

All the experiments in this paper were performed in Matlab on a Pentium PC with 1.73 GHz CPU and 4G RAM. For the audio-waveform example, 80 VB iterations for LSBP required 40 seconds. For the multi-task image segmentation, H-LSBP required nearly 7 hours of CPU to jointly segment 300 images, using 60 VB iterations (this CPU time may be cut in half if we only use 30 VB iterations, with minor degradation in performance). With both experiments, KSBP/H-KSBP typically required comparable CPU time, while DP/HDP required less than half the CPU time.

6.5 Summary

We propose the *hierarchical* logistic stick-breaking process, H-LSBP, to segment multiple data sets simultaneously, with example results presented for images. The model parameters (atoms) are shared across all images, using a shared draw from a global DP prior. The total number of important atoms across all images, as well as the particular important atoms for a specific image, are inferred with an efficient variational Bayesian (VB) solution. Compared with the hierarchical Dirichlet process (HDP) and the hierarchical KSBP, the proposed method yields superior segmentation performance, based on studies with natural images. Further, we have investigated the ability of HDP, H-KSBP and H-LSBP to infer inter-relationship between different images, based on the underlying sharing of model atoms. The improved segmentation quality of the H-LSBP, relative to HDP and H-KSBP, also yields an improved ability to infer inter-image relationships.

Concerning future research, the results in Figure 6.4 indicate that the inferred atoms have connections to physical entities in images. This suggests that the model may be extended to the joint modeling of images and text [BDF$^+$03], with the text as-

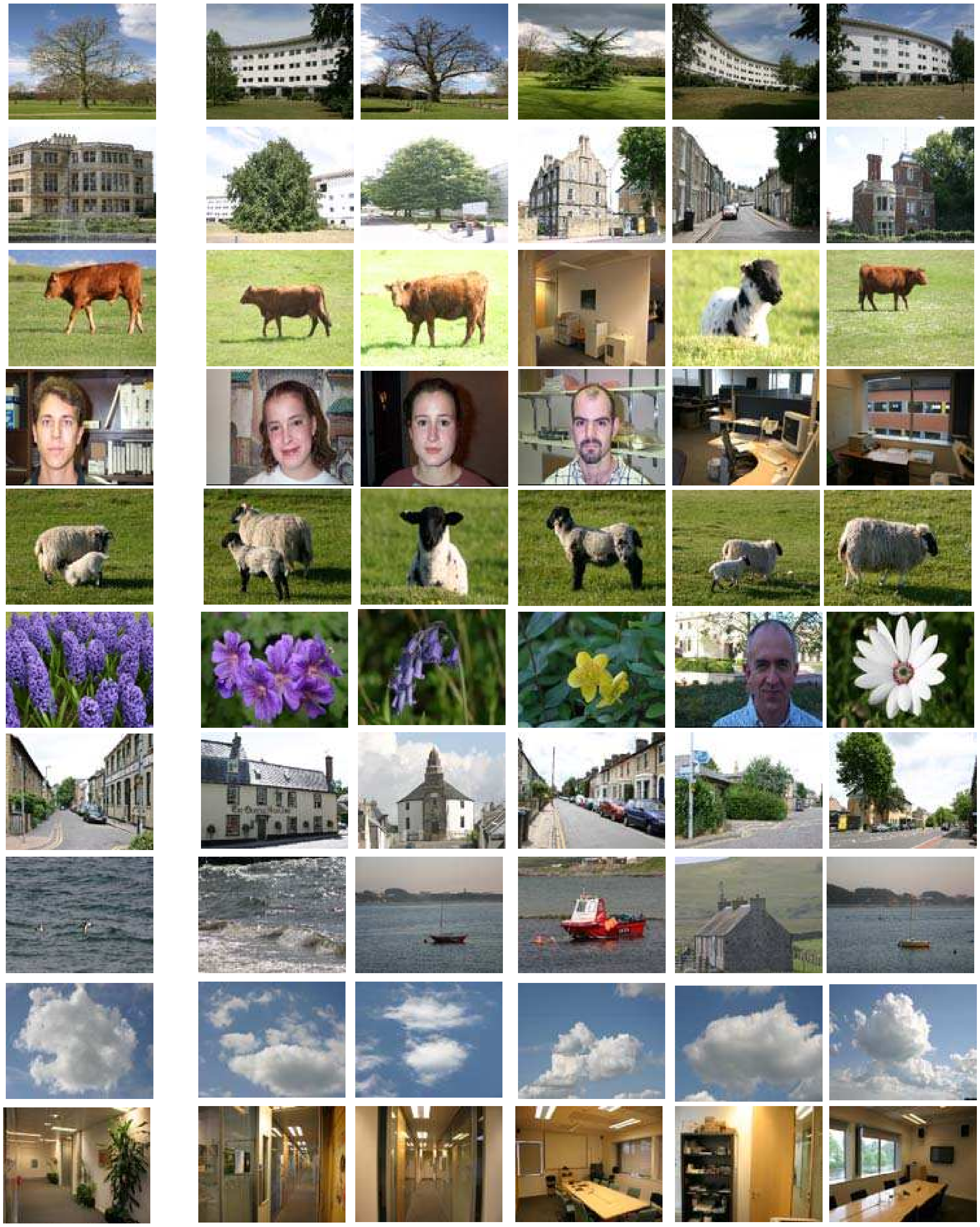


FIGURE 6.8: Sample image sorting result, as generated by H-LSBP. The first left column shows the images inquired, followed by the five most similar images from the second to sixth column.

sociated with aspects of the image. In addition, in the H-LSBP modeling of multiple images, the employed DP prior assumes that the order of the images is exchangeable (although LSBP imposes that spatial location within a particular image is not exchangeable). There are many applications (*e.g.*, video) for which the multiple images may have a prescribed time index, that should be exploited. The results on the time-dependent audio data demonstrate how LSBP may also be employed to exploit temporal information.

7

Conclusions and Future Work

Temporal and spatial information are usually included within data, removing the exchangeability assumption underlying many nonparametric Bayesian models. Such information may be exploited to help infer patterns, clusters or segments in the data. In recent years, nonparametric techniques have been widely employed for clustering data, and the number of clusters is inferred automatically in a data-driven way. Unfortunately most of these methods do not explicitly exploit the appended order information and assume all observations are exchangeable. Although this assumption usually leads to a tractable model form for the posterior computation, it may degrade the clustering performance as the data is exposed to noise.

To address the challenges, this thesis considered this problem into two parts: (i) evolutionary clustering of time-evolving data, which assumes the temporal order existed across different groups while allows the data exchangeable within each group. This is distinct with the traditional Markov model assumption, as it considers to capture a long temporal dependence associated with data. Additionally, (ii) segmentation of spatially-dependent data has been considered, with which the order cannot be naturally defined but the relative position encodes the “closeness” between each two local sites. This will be helpful to infer the cluster number underlying the data, as the spatial dependence promotes a compact model solution.

Based on these considerations, several innovative work have been addressed in this thesis, summarized as follows:

- The dynamic hierarchical Dirichlet process (dHDP) is proposed to model the time-evolving statistical properties of sequential data sets. The data collected at any time point are represented via a mixture associated with an appropriate underlying model, in the framework of HDP. The statistical properties of data collected at consecutive time points are linked via a random parameter that controls their probabilistic similarity. The sharing mechanisms of the time-evolving data are derived, and a relatively simple Markov Chain Monte Carlo sampler is developed.

- A dynamic hidden Markov model (HMM) mixture is developed to model complex sequential data, with a focus on audio signals from music. The music is represented in terms of a sequence of discrete observations, and the sequence is modeled using a hidden Markov model (HMM) with time-evolving parameters. Segmentation of a given musical piece is constituted via the model inference. Detailed examples are presented on several pieces, with comparisons to other models. The experimental results are also compared with a conventional music-theoretic analysis.

- A logistic stick-breaking process (LSBP) is proposed for non-parametric clustering of general spatially- or temporally-dependent data, imposing the belief that proximate data are more likely to be clustered together. The sticks in the LSBP are realized via multiple logistic regression functions, with shrinkage priors employed to favor contiguous and spatially localized segments. Efficient variational Bayesian inference is derived, and comparisons are made to related techniques in the literature. Experimental analysis is performed for audio waveforms and images, and it is demonstrated that for segmentation applications the LSBP yields generally homogeneous segments with sharp boundaries.

- The LSBP is also extended for the simultaneous processing of multiple data sets, yielding a hierarchical logistic stick-breaking process (H-LSBP). The model parameters (atoms) within the H-LSBP are shared across the multiple learning tasks. Exploiting the information contained in multiple data sets (tasks), model-parameter estimation may be improved and the sharing mechanism is also inferred as a posterior. We employ the H-LSBP to simultaneously segment multiple images. In addition to inferring a segmentation of each image, the framework allows sorting and searching among the images.

These contributions motivate several directions for future work:

1. Exploit the temporal or spatial information for a classification problem. In the thesis, we focused on clustering temporally or spatially dependent data. However, in some cases, labeled data are available at each time-stamp or a local neighborhood. It is promising to assume that proximate observations are more likely to share the same classification boundary, hence yielding a mixture of experts dependent with each other. Including the prior information into designing classifiers, it may improve classification rate especially for the missing data in semisupervised learning, or be helpful for prediction of the incoming data.

2. Learn patterns for the data with both temporally and spatially dependence. For joint segmenting multiple images, H-LSBP is proposed to share the mixture of atoms globally across different tasks. Although spatially dependence is modeled for each task with an LSBP, different tasks are still assumed to be exchangeable. This is not always true for specific applications, such as video analysis. In these spatio-temporal data set, different data-dependent structures exist along time and

within 2-D space respectively. Therefore, a more general framework is necessary to considered for joint learning temporally- and spatially- dependent patterns. For example, each spatially dependent task is modeled by a mixture model with LSBP and the temporal dependence associated with different tasks may be constructed via a Gaussian process.

3. A spatially-dependent mixture model has been developed for image segmentation in this thesis. However, feature extraction, as a preprocess for segmentation, is a trivial but time-consuming work. More importantly, the feature extraction may considerably influence the segmentation performance, no matter which model is used for the clustering. Accordingly, it will be more interesting to build a spatially-dependent model generating images directly instead of generating appearance features. Motivated by this, we might extend the dependent model structures introduced here to a non-parametric Bayesian dictionary learning problem [ZCP+09], so that proximate image patches are generated from a similar factor-combination. As a result, smooth segmentation and feature extraction will be finished simultaneously.

Appendix A

Proof of Theorem 2

According to (3.6), $G_j = w_{j1}G_1 + \sum_{l=2}^{j} w_{jl}H_{l-1}$, where $w_{jl} = \tilde{w}_{l-1}\prod_{m=l}^{j-1}(1-\tilde{w}_m)$. Then given $\{w_{j1}, \ldots, w_{jj}\}$ and the base distribution H, the expectation of G_j is

$$\begin{aligned} E\big\{G_j(B)\big\} &= w_{j1}E\big\{G_1(B)\big\} + \sum_{l=2}^{j} w_{jl}E\big\{H_{l-1}(B)\big\} \\ &= \sum_{l=1}^{j} w_{jl}H(B). \end{aligned} \tag{A.1}$$

Since given G_0, the variance of $G_j(B)$ is $V\big\{G_j(B)|G_0(B)\big\} = \sum_{l=1}^{j}(\frac{w_{jl}^2}{\alpha_{0l}+1})G_0(B)\big\{1-G_0(B)\big\}$. Then we can get the variance of $G_j(B)$ with the expectation of $G_0(B)$ as follows:

$$
\begin{aligned}
V\big\{G_j(B)\big\} &= E\Big[V\big(G_j(B)|G_0(B)\big)\Big] + V\Big[E\big(G_j(B)|G_0(B)\big)\Big] \\
&= E\Big[\sum_{l=1}^{j}\Big(\frac{w_{jl}^2}{\alpha_{0l}+1}\Big)G_0(B)\big(1-G_0(B)\big)\Big] + V\Big[\sum_{l=1}^{j} w_{jl}G_0(B)\Big] \\
&= \sum_{l=1}^{j}\frac{w_{jl}^2}{\alpha_{0l}+1}E\Big[G_0(B)-G_0^2(B)\Big] + V\Big[\sum_{l=1}^{j} w_{jl}G_0(B)\Big] \\
&= \sum_{l=1}^{j}\frac{w_{jl}^2}{\alpha_{0l}+1}\Big[H(B)-\big(V(G_0(B))+H^2(B)\big)\Big] + \sum_{l=1}^{j} w_{jl}^2 V\big[G_0(B)\big] \\
&= \sum_{l=1}^{j} w_{jl}^2\Big[\big(1-\frac{1}{1+\alpha_{0l}}\big)V\big[G_0(B)\big] + \frac{H(B)\big[1-H(B)\big]}{1+\alpha_{0l}}\Big] \\
&= \sum_{l=1}^{j} w_{jl}^2\Big[\frac{\alpha_{0l}}{1+\alpha_{0l}}\cdot\frac{1}{1+\gamma}H(B)\big[1-H(B)\big] + \frac{H(B)\big[1-H(B)\big]}{1+\alpha_{0l}}\Big] \\
&= \sum_{l=1}^{j}\frac{w_{jl}^2}{1+\alpha_{0l}}\big(\frac{\alpha_{0l}+\gamma+1}{1+\gamma}\big)H(B)\big[1-H(B)\big].
\end{aligned}
\tag{A.2}
$$

Additionally given G_0 we can get

$$
\begin{aligned}
&E\{G_j(B)G_{j-1}(B)\} - E\{G_j(B)\}E\{G_{j-1}(B)\} \\
=&E\Big[\{w_{j1}G_1(B)+\ldots+w_{jj}H_{j-1}(B)\}\{w_{j-1,1}G_1(B)+\ldots+w_{j-1,j-1}H_{j-2}(B)\}\Big] \\
&-E\{w_{j1}G_1(B)+\ldots+w_{jj}H_{j-1}(B)\}E\{w_{j-1,1}G_1(B)+\ldots+w_{j-1,j-1}H_{j-2}(B)\} \\
=&w_{j1}w_{j-1,1}V\{G_1(B)\} + \sum_{l=2}^{j-1} w_{jl}w_{j-1,l}V\{H_{l-1}(B)\} \\
=&\sum_{l=1}^{j-1}\frac{w_{jl}w_{j-1,l}}{1+\alpha_{0l}}\cdot\frac{\alpha_{0l}+\gamma+1}{1+\gamma}H(B)\big[1-H(B)\big].
\end{aligned}
\tag{A.3}
$$

From the above analysis, the correlation coefficient of the distributions between the adjacent groups defined in (3.11) can be formularized as follows:

$$Corr(G_{j-1}(B), G_j(B)) = \frac{\sum_{l=1}^{j-1} \frac{w_{jl} w_{j-1,l}}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)}{\left[\sum_{l=1}^{j} \frac{w_{jl}^2}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)\right]^{1/2} \left[\sum_{l=1}^{j-1} \frac{w_{j-1,l}^2}{1+\alpha_{0l}} \cdot (\alpha_{0l} + \gamma + 1)\right]^{1/2}}. \tag{A.4}$$

Appendix B

Posterior Update for the Hidden Markov Models (HMM)

Dirichlet distributions are used as the conjugate priors for these parameters [QPC07]. Assume I, M respectively represent the number of states for each HMM and the codebook size of the discrete sequential observations, then $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{q}$ in expression (B.1) are chosen as $\mathbf{u} = 1/I\mathbf{1}_I$, $\mathbf{v} = 1/M\mathbf{1}_M$ and $\mathbf{q} = 1/I\mathbf{1}_I$, with $\mathbf{1}_a$ denoting an $a \times 1$ vector of ones. These priors are chosen motivated by the results of [IZ02], which imply that each of the probability vectors converge to a Dirichlet process with precision parameter one, as the number of elements increases. The Dirichlet process is appealing in favoring a few dominant elements, with the remaining probabilities close to zero.

$$
\begin{aligned}
Pr(\mathbf{A}_k^*|\mathbf{u}) &= \prod_{\rho=1}^{I} Dir\Big(\{a_{\rho 1}, \ldots, a_{\rho I}\}; \mathbf{u}\Big) \\
Pr(\mathbf{B}_k^*|\mathbf{v}) &= \prod_{\rho=1}^{I} Dir\Big(\{b_{\rho 1}, \ldots, b_{\rho M}\}; \mathbf{v}\Big) \qquad \text{(B.1)} \\
Pr(\boldsymbol{\pi}_k^*|\mathbf{q}) &= Dir\Big(\{\pi_1, \ldots, \pi_I\}; \mathbf{q}\Big)
\end{aligned}
$$

where I is the number of states for the HMM and M is the codebook size of the observations. To update the parameters of HMMs, we also sample the hidden states. The posterior updating equations are as follows:

- $Pr(\mathbf{A}_k^*|\mathbf{A}_{-k}^*, \mathbf{B}^*, \boldsymbol{\pi}^*, \mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{\alpha}_0, \mathbf{s}, \mathbf{z}, \mathbf{r}, \mathbf{x})$

$= \prod_{\rho=1}^{I} Dir\Big(\{a_{\rho 1}, \ldots, a_{\rho I}\}; \{u_1 + \hat{u}_{k,\rho 1}, \ldots, u_I + \hat{u}_{k,\rho I}\}\Big)$

where $\hat{u}_{k,\rho\xi} = \sum_{j=1:z_{jk}=1}^{J} \sum_{t=1}^{T-1} \delta(s_{j,k,t} = \rho, s_{j,k,t+1} = \xi)$.

- $Pr(\mathbf{B}_k^*|\mathbf{A}^*, \mathbf{B}_{-k}^*, \boldsymbol{\pi}^*, \mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{\alpha}_0, \mathbf{s}, \mathbf{z}, \mathbf{r}, \mathbf{x})$

$= \prod_{\rho=1}^{I} Dir\Big(\{b_{\rho 1}, \ldots, b_{\rho M}\}; \{v_1 + \hat{v}_{k,\rho 1}, \ldots, v_M + \hat{v}_{k,\rho M}\}\Big)$

where $\hat{v}_{k,\rho m} = \sum_{j=1:z_{jk}=1}^{J} \sum_{t=1}^{T} \delta(s_{j,k,t} = \rho, x_{jt} = m)$.

- $Pr(\boldsymbol{\pi}_k^*|\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{\pi}_{-k}^*, \mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{\alpha}_0, \mathbf{s}, \mathbf{z}, \mathbf{r}, \mathbf{x})$

$= Dir\Big(\{\pi_1, \ldots, \pi_I\}; \{q_1 + \hat{q}_{k,1}, \ldots, q_I + \hat{q}_{k,I}\}\Big)$ where $\hat{q}_{k,\rho} = \sum_{j=1:z_{jk}=1}^{J} \delta(s_{j,k,1} = \rho)$.

- $Pr(\mathbf{s}_{j,k,t}|\mathbf{s}_{-j}, \mathbf{s}_{j,k,-t}, \mathbf{A}^*, \mathbf{B}^*, \boldsymbol{\pi}^*, \mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{\alpha}_0, \mathbf{z}, \mathbf{r}, \mathbf{x})$

$$\propto \left\{ \begin{array}{lll} Pr(s_{j,k,1}) Pr(s_{j,k,2}|s_{j,k,1}) Pr(x_{j1}|s_{j,k,1}) & for & t = 1 \\ Pr(s_{j,k,t}|s_{j,k,t-1}) Pr(s_{j,k,t+1}|s_{j,k,t}) Pr(x_{jt}|s_{j,k,t}) & for & 2 \le t \le T-1 \\ Pr(s_{j,k,T}|s_{j,k,T-1}) Pr(x_{jT}|s_{j,k,T}) & for & t = T \end{array} \right.$$

Bibliography

[AE06] A. Argyrion and T. Evgeniou. Multi-task feature learning. *In Proc. of Neural Information Processing Systems*, 2006.

[Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[Ald85] D. Aldous. Exchangeability and related topics. *École d'été de probabilités de Saint-Flour*, XIII-1983:1–198, 1985.

[Ant74] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.*, 2(6):1152–1174, 1974.

[AP02] J. Aucouturier and F. Pachet. Music similarity measures: what's the use ? *In Proc. of the International Symposium on Music Information Retrieval*, 2002.

[AP09] T. Ahonen and M. Pietikäinen. Image discription using joint distribution of filter bank responses. *Pattern Recognition Letters*, 30:368–376, 2009.

[AWS$^+$08] Q. An, C. Wang, I. Shterev, E. Wang, L. Carin, and D. B. Dunson. Hierarchical kernel stick-breaking process for multi-task image analysis. In *Proceedings of the International Conference of Machine Learning*, 2008.

[AX08] A. Ahmed and E. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. *In Proc. of the 8th SIAM International Conference on Data Mining*, pages 219–230, 2008.

[AZ05] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, November 2005.

[BAW07] E. V. Bonilla, F. V. Agakov, and C. K. I. Williams. Kernel multi-task learning using task-specific features. *In Proc. of the 11th AISTATS*, 2007.

[BBBG04] M. Ben, M. Betser, F. Bimbot, and G. Gravier. Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs. In *Proceedings of the International Conference on Spoken Language Processing*, 2004.

[BCG03] S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC, 1st edition, 2003.

[BDF$^+$03] K. Barnard, P. Duygulu, D. Forsyth, N.D. Freitas, D.M. Blei, and M.I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[BDS03] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. *In Proc. of Computational Learning Theory*, 2003.

[BE67] 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.*, 73:360–363, 1967.

[Bea03] M. J. Beal. *Variational algorithms for approximate Bayesian inference.* PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

[BF05] P. Boyle and M. Frean. Dependent Gaussian processes. In *In Proc. of Advances in Neural Information Processing Systems*, 2005.

[BGFS08] S. Banerjee, A. E. Gelfand, A. O. Finley, and H. Sang. Gaussian predictive process models for large spatial data sets. *Jorunal of the Royal Statistical Society, Series B*, 70(4):825–848, 2008.

[BGJT04] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems.* MIT Press, 2004.

[BGR02] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. *Advances in Neural Information Processing Systems*, 2002.

[BH03] B. Bakker and T. Heskes. Task clustering and gating for Bayeian multitask learning. *Journal of Machine Learning Research*, 4:83–699, May 2003.

[BJ04] D. M. Blei and M. I. Jordan. Variational methods for the Dirichlet process. In *In Proc. of the International Conference on Machine Learning*, 2004.

[BL06] D.M. Blei and J.D. Lafferty. Dynamic topic models. In *Proceedings of the International Conference on Machine Learning*. Pittsburgh, USA, 2006.

[BM73] D. Blackwell and J.B. MacQueen. Ferguson distributions via Polya urn schemes. *Ann. Statist.*, 1(2):353–355, 1973.

[BM96] C.A. Bush and S.N. MacEachern. A semiparametric Bayesian model for randomised block designs. *Biometrika*, 83(2):275–285, 1996.

[BNJL03] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[BP66] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.*, 37:1554–1563, 1966.

[BPSW] 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(1).

[BS03] C. M. Bishop and M. Svensén. Bayesian hierarchical mixture of experts. In *Proceedings of the Ninetheenth Conference of Uncertainty in Artificial Intelligence*, 2003.

[BT00] C. M. Bishop and M. E. Tipping. Variational relevance vector machines. In *Proceedings of the sixth Conference of Uncertainty in Artificial Intelligence*, 2000.

[Bur03] C. Burkhart. *Anthology for Music Analysis*. Schirmer Bookds, New York, USA, 2003.

[BVZ98] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[Car97] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[CBS05] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[CD09a] Y. Chung and D. B. Dunson. Nonparametric Bayes conditional distribution modeling with variable selection. *Journal of the American Statistical Association*, 2009.

[CD09b] Y. Chung and D.B. Dunson. The local Dirichlet process. *Annals of the Institute of Statistical Mathematics*, Online first:Springer Netherlands, 2009.

[CDD07] F. Caron, M. Davy, and A. Doucet. Generalized Polya Urn for time-varying Dirichlet process mixtures. In *In Proc. of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 33–40, Corvallis, Oregon, 2007. AUAI Press.

[CDD$^+$08] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe. Bayesian inference for linear dynamic models with Dirichlet process mixtures. *IEEE Transactions on Signal Processing*, 56:71–84, 2008.

[CFF07] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentatoin and classification of objects and scenes. *In IEEE Conf. on Computer Vision (ICCV)*, 2007.

[CG05] W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. *In Proc. of International Conference of Machine Learning*, 2005.

[DC04] H. Deng and D. A. Clausi. Unsupervised image segmentation using a simple MRF model with a new implementation scheme. *Pattern Recognition*, 37:2323–2335, 2004.

[DGG07] J. A. Duan, M. Guindani, and A. E. Gelfand. Generalized spatial Dirichlet process models. *Biometrika*, 94(4):809–825, 2007.

[DP07] D. B. Dunson and J. Park. Kernel stick-breaking processes. *Biometrika*, 95:307–323, 2007.

[Dun06] D.B. Dunson. Bayesian dynamic modeling of latent trait distributions. *Biostatistics*, 7(4):551–568, 2006.

[EW95] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[FH04] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[FSJW08a] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. The sticky HDP-HMM for systems with state persistence. In *Proceedings of the International Conference on Machine Learning*, 2008.

[FSJW08b] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. An HDP-HMM for systems with state persistence. In *In Proc. of the 25th International Conference on Machine Learning*. Helsinki, Finland, 2008.

[GFK05] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the International Conference on Speech and Computer*, 2005.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[GG92] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Springer, NewYork, USA, 1992.

[GKM05] A. E. Gelfand, A. Kottas, and S. N. Maceachern. Bayesian nonparametric spatial modeling with Dirichlet process mixing. *Journal of the American Statistical Association*, 100(471):1021–1035, 2005.

[GLSGFV06] P. J. García-Laencina, J. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing values using multitask learning. *In Proc. of International Joint Conference on Neural Networks*, 2006.

[GRS98] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1st edition, 1998.

[GS04] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101, Suppl 1:5228–5235, 2004.

[GS06] J.E. Griffin and M.F.J. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101:179–194, 2006.

[GWP08] Ryan Gomes, Max Welling, and Pietro Perona. Memory bounded inference in topic models. In *Proceedings of the International Conference of Machine Learning*, 2008.

[Hoc01] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov random fields and related problems. *Journal of the ACM*, 48(4):686–701, 2001.

[IJ01] H. Ishwaran and L.F. James. Gibbs sampling methods for Stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[IMRM04] M. De Iorio, P. Mű̈ller, G. L. Rosner, and S. N. MacEachern. An ANOVA model for dependent random measures. *Journal of the American Statistical Association*, 99:205–215, 2004.

[IR05] H. Ishwaran and J. S. Rao. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730C773, 2005.

[IZ02] H. Ishwaran and M. Zarepour. Exact and approximate sum-representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.

[JCMJ06] J.H. Jensen, M.G. Christensen, M.N. Murthi, and S.H. Jensen. Evaluation of MFCC estimation techniques for music similarity. In *Proceedings of the 14th European Signal Processing.* Florence, Italy, 2006.

[JG09] G. Jun and J. Ghosh. Spatially adaptive classification and active learning of multispectral data with Gaussian processes. In *In Proc. of IEEE International Conference on Data Mining Workshops*, 2009.

[KMH05] H. Kim, B. K. Mallick, and C. C. Holmes. Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100:653–668, 2005.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *In Proc. of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[KP06] Z. Kato and T. Pong. A Markov random field image segmentation model for color textured images. *Image and Vision Computing*, 24:1103C1114, 2006.

[KS80] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications.* AMS, 1980.

[LI09] J. Luttinen and A. Ilin. Variational Gaussian-process factor analysis for modeling spatio-temporal data. In *In Proc. of Advances in Neural Information Processing Systems*, 2009.

[LJ01] Q. Lu and T. Jiang. Pixon-based image denoising with Markov random fields. *Pattern Recognit*, 34:2029–2039, 2001.

[LLC07] Q. Liu, X. Liao, and L. Carin. Semi-supervised mulitask learning. *In Proc. of Neural Information Processing Systems*, 2007.

[Log00] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*. Plymouth, USA, 2000.

[Mac99] S. N. MacEachern. Dependent nonparametric processes. *In ASA Proceedings of the Section on Bayesian Statistical Science*, 1999.

[Mac00] S. N. MacEachern. Dependent Dirichlet processes. Technical report, 2000.

[Mei03] M. Meilă. Comparing clusterings by the variation of information. In *Proceedings of the Sixteenth Annual Conference ofn Computational Learning Theory*, 2003.

[Mor05] G. Mori. Guiding model search using segmentation. In *Proceedings of the International Conference on Computer Vision*, 2005.

[MP04] C. A. Micchelli and M. Pontil. Kernels for multi-task learning. *In Proc. of Neural Information Processing Systems*, 2004.

[MS02] M. Medvedovic and S. Sivaganesan. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18(9):1194–1206, 2002.

[NCD07] K. Ni, L. Carin, and D. B. Dunson. Multi-task learning for sequential data via iHMMs and the nested Dirichlet process. *In Proc. of the International Conference on Machine Learning*, 2007.

[NG08] X. Nguyen and A. E. Gelfand. The Dirichlet labeling process for functional data analysis. Technical Report T.R. 08-37, Dept. of Statistical Science, Duke University, 2008.

[NPCD08] K. Ni, J. Paisley, L. Carin, and D. Dunson. Multi-task learning for analyzing and sorting large databases of sequential data. *IEEE Trans. Signal Processing*, 56:3918–3931, 2008.

[OB06] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric Bayesian inference. In *In Proc. of European Conference on Computer Vision*, 2006.

[OB08] P. Orbanz and J. M. Buhmann. Nonparametric Bayesian image segmentation. *International Journal of Computer Vision*, 77:25–45, 2008.

[PD09] J. Park and D. B. Dunson. Bayesian generalized product partition model. *Statistica Sinica*, 2009.

[PFH$^+$06] T. Plotz, G.A. Fink, P. Husemann, S. Kanies, K. Lienemann, T. Marschall, M. Martin, L. Schillingmann, M. Steinrucken, and H. Sudek. Automatic detection of song changes in music mixes using stochastic models. In *18th International Conference on Pattern Recognition (ICPR'06)*. Hong Kong, China, 2006.

[PGBE07] J.-F. Paiement, Y. Grandvalet, S. Bengio, and D. Eck. A generative model for rhythms. In *NIPS'2007 Music, Brain & Cognition Workshop*. Whistler, Canada, 2007.

[PGG09] S. Petrone, M. Guindani, and A. E. Gelfand. Hybrid Dirichlet mixture models for functional data. *Journal Royal Statistical Society, Ser. B*, 2009.

[PSH08] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. *In ECCV*, 2008.

[QPC07] Y. Qi, J. W. Paisley, and L. Carin. Music analysis using hidden Markov mixture models. *IEEE Transactions on Signal Processing*, 55:5209–5224, 2007.

[Rab89] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Rap99] C. Raphael. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

[Ras00] C. E. Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 2000.

[RBkT$^+$05] N. Ramakrishnan, C. Bailey-kellogg, S. Tadepalli, V. N. Pandey, and V. N. P. Gaussian processes for active data mining of spatial aggregates. In *In Proceedings of the SIAM International Conference on Data Mining*, 2005.

[RD09] A. Rodriguez and D. B. Dunson. Nonparametric Bayesian models through probit stick-breaking processes. Submitted to *Journal of the Royal Statistical Society, Series B*, 2009.

[RDG08] A. Rodríguez, D. B. Dunson, and A. E. Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103(483), 2008.

[RH08] A. Rodriguez and E. Ter Horst. Dynamic density estimation with financial applications. *Bayesian Analysis*, 3:339–366, 2008.

[RI09] P. Rai and H. Daumé III. Multitask learning using nonparametrically learned predictor subspaces. 2009.

[RL92] A.E. Raftery and S. Lewis. How many iterations in the Gibbs sampler? *Bayesian Stat.*, 4:763–773, 1992.

[RM03] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision*, 2003.

[RW06] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[Sch78] G. E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–462, 1978.

[Set94] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 2:639–650, 1994.

[She08] Daniel Sheldon. Graphical multi-task learning. In NIPS 2008 Workshop on Structured Input and Structured Output, 2008.

[SJ08] E. B. Sudderth and M. I. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *Proceedings of the Neural Information Processing Systems*, 2008.

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[SNG07] G. Sfikas, C. Nikou, and N. P. Galatsanos. Robust image segmentation with mixtures of Student's t-distributions. In *Proceedings of the IEEE International conference on Image Processing*, 2007.

[SNP08] G. Sfikas, C. Nikou, and N. P.Galatsanos. Edge preserving spatially varying mixtures for image segmentaiton. In *Proceedings of the International conference on Computer Vision and Pattern recognition*, 2008.

[Ste99] M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer, 1999.

[SYL$^+$09] M. R. Sabuncu, B. T. Thomas Yeo, K. V. Leemput, B. Fischl, and P. Golland. Nonparametric mixture models for supervised image parcellation. In *In Proc. MICCAI Workshop on Probabilistic Models for Medical Image Analysis*, 2009.

[Tem07] D. Temperley. *Music and Probability.* MIT Press, Cambridge, Massachusetts (USA), 2007.

[Tem08] D. Temperley. A probabilistic model of melody perception. *Cognitive Science*, 32:418–444, 2008.

[Tip01] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research 1*, pages 211–244, 2001.

[TJBB06] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[TK03] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press; 2 edition, 2003.

[TO96] S. Thrun and O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. *In Proc. of International Conference of Machine Learning*, 1996.

[TR06] S. Tranter and D. Reynolds. An overview of automatic speaker diarisation systems. *IEEE Trans. on Audio, Speech, and Language Processing*, 14:1557–1565, 2006.

[UPH07] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):929–944, 2007.

[VZ02] M. Varma and A. Zisserman. Classifying images of materials: achieving viewpoint and illumination independence. *In Proceedings of the 7th European Conference on Computer Vision*, 2002.

[WB05] J. Winn and C. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

[WG07] X. Wang and E. Grimson. Spatial latent Dirichlet allocation. *In Proceedings of Neural Information Processing Systems Conference*, 2007.

[WS06] J. V. D. Weijer and C. Schmid. Coloring local feature extraction. In *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[WT90] G. C. G. Wei and M. A. Tanner. A monte Carlo implementation of the EM algorithm and the Poor Mans data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[XLC07] Y. Xue, X. Liao, and L. Carin. Multi-task learning for classifiction with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, January 2007.

[YTS05] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. *In Proc. of International Conference of Machine Learning*, 2005.

[ZCP+09] M. Zhou, H. Chen, J. W. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric Bayesian dictionary learning for sparse image representations. *In Proc. of Advances in Neural Information Processing Systems*, 2009.

[ZK04] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Proceedings of CVPR'04*, 2004.

Biography

Lu Ren was born on Feb, 1980 in Xi'an, China. She received her B.S. degree in Electrical Engineering and M.S. degree in Signal Processing from Xidian University, Xi'an, China. Beginning August 2005, she began to study in the department of Electrical and Computer Engineering at Duke University, where she received the M.S. degree and Ph.D degree in 2007 and 2010 respectively. Her current research interests include machine learning, Bayesian statistics, computer vision and signal processing.

Publications

L. Ren, D. B. Dunson and L. Carin (2008). The Dynamic Hierarchical Dirichlet Process, International Conference on Machine Learning (ICML) 2008, Helsinki, Finland.

L. Ren, D. B. Dunson, S. Lindroth and L. Carin (2009). Dynamic Nonparametric Bayesian Models for Analysis of Music, to appear in J. American Statistical Association (JASA).

L. Ren, D. B. Dunson, S. Lindroth and L. Carin (2009). Music Analysis with a Bayesian Dynamic Model, IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) 2009, Taipei, Taiwan.

L. Ren, L. Du, L. Carin and D. Dunson (2009). Logistic Stick-Breaking Process, submitted to J. Machine Learning Research.

I. Pruteanu-Malinici, L. Ren, J. Paisley, E. Wang and L. Carin (2009), Hierarchical Bayesian Modeling of Topics in Time-Stamped Documents, to appear in IEEE Trans. Pattern Analysis Machine Intelligence.

L. Du, L. Ren, D. Dunson and L. Carin(2009), A Bayesian Model for Simultaneous Image Clustering, Annotation and Object Segmentation, to appear in Neural Information Processing Systems(NIPS) 2009, Vancouver, Canada.

M. Zhou, H. Chen, J. Paisley, L. Ren and L. Carin (2009), Non-Parametric Bayeisan Dictionary Learning for Sparse Image Representations, to appear in Neural Information Processing Systems (NIPS) 2009, Vancouver, Canada.

L. Ren, M. Xing and Z. Bao (2005), Adaptive despeckling SAR images based on scale space correlation, IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) 2005, Philadelphia, USA.