Bayesian Nonparametric Discovery of Layers and Parts from Scenes and Objects

by

Soumya Ghosh

B. E., University of Mumbai, 2004

M. S., University of Houston, 2006

M. S., University of Colorado, 2009

A Dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Computer Science at Brown University

Providence, Rhode Island
June 2015

This dissertation by Soumya Ghosh is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date ______________ ____

Erik B. Sudderth, Advisor

Recommended to the Graduate Council

Date ______________ ____

Michael J. Black, Reader

Date ______________ ____

James Hays, Reader

Approved by the Graduate Council

Date ______________ ____

Peter M. Weber,
Dean of the Graduate School

Vitae

Soumya Ghosh was born on June 11, 1983 in Mumbai, India.

Education

- Ph.D. in Computer Science, Brown University, Providence, RI, May 2015.
- M.Sc. in Computer Science, University of Colorado, Boulder, CO, May 2009.
- M.Sc. in Computer Science, University of Houston, Houston, TX, Dec 2006.
- B.E. in Computer Engineering, University of Mumbai, Mumbai, India May 2004.

Publications

Journal Articles

- Soumya Ghosh, Tomasz F. Stepinski, Ricardo Vilalta. Automatic Annotation of Planetary Surfaces With Geomorphic Labels. *IEEE Transactions on Geoscience and Remote Sensing*, 48, 175–185, 2010.

Refereed Conference Proceedings

- Soumya Ghosh, Michalis Raptis, Leonid Sigal, Erik Sudderth. “Nonparametric Clustering with Distance Dependent Hierarchies.” *30th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- Soumya Ghosh, Erik Sudderth, Matthew Loper, Michael J. Black. From “Deformations to Parts: Motion-based Segmentation of 3D Objects.” *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.
- Soumya Ghosh, Erik Sudderth. “Nonparametric learning for layered segmentation of natural images.” *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- Soumya Ghosh, Andrei B. Ungureanu, Erik Sudderth, David Blei. “Spatial distance dependent Chinese restaurant processes for image segmentation.” *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011.

- Soumya Ghosh, Jane Mulligan. "A segmentation guided label propagation scheme for autonomous navigation." *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- Soumya Ghosh, Joseph Pfeiffer III, Jane Mulligan. "A general framework for reconciling multiple weak segmentations of an image." *Workshop on Applications of Computer Vision (WACV)*, 2009.
- Steven Bethard, Soumya Ghosh, James Martin, Tamara Sumner. Topic Model "Methods for Automatically Identifying Out-of-Scope Resources." *Joint Conference on Digital Libraries (JCDL)*,2009.
- Soumya Ghosh, Soundar Srinivasan, Burt Andrews. "Using Weak Supervision in Learning Gaussian Mixture Models." *International Joint Conference on Neural Networks (IJCNN)*, 2009.
- Tomasz F. Stepinski, Soumya Ghosh, Ricardo Vilalta. "Machine learning for automatic mapping of planetary surfaces." *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2007.

Invited Papers

- Tomasz F. Stepinski, Ricardo Vilalta and Soumya Ghosh. "Machine Learning Tools for Automatic Mapping of Martian Landforms." *IEEE Intelligent Systems* 22, 6, 100–106, Nov 2007.

Acknowledgements

This thesis would not have been possible without the constant encouragement and seemingly endless patience of Professor Erik B. Sudderth. I joined Erik's fledgling research group during my first semester at Brown, and have benefitted from his clever ideas, rigorous methodical approach to challenging problems and careful mentorship, ever since. His many insightful suggestions that elegantly simplified complex technical challenges, have immensely improved the research detailed in this thesis. I owe a great deal of my achievements over the past six years to Erik.

I would like to thank my committee members Professors Michael J. Black and James Hays for their thoughtful comments, suggestions and mentorship. One of my first courses at Brown was Michael's brilliantly taught computer vision class. It reinvigorated my interest in the area and engendered many of the ideas developed in this thesis. I have also had the good fortune of spending two productive summers working with incredible mentors, Dr. Ce Liu and Professor Yair Weiss at MSR and Doctors Leon Sigal and Michalis Raptis at Disney research.

A special thanks goes out to Mike Hughes, Dae Il Kim, Jason Pacheko, and Ben Swanson for being great lab mates and even better friends and confidants. Without you all, the difficult times would have been unbearable and the good times would not have been nearly as fun. Thank you! I would also like to thank many past and present graduate students at Brown, Jesse Butterfield, Carleton Coffrin, Micha Elsner, Peng Guan, Geng Ji, Layla Oesper, Genevieve Patterson, Zhile Ren, Anna Ritz, Eric Sodomka, Deqing Sun, Geoff Sun, Aggleki Tsoli and Silvia Zuffi for innumerable interesting conversations and discussions over the years.

Brown has been a very conducive environment for my research. The incredible resources provided by the computer science department have been vital to the success of this thesis. I am particularly grateful to Lauren Clarke. I shudder to imagine grad life without her constant help navigating the bureaucratic maze.

I am forever grateful to my parents for their unconditional love and support in all my endeavors. Without their unwavering patience and understanding, I would not have been able to freely pursue my academic interests. Finally, I would be remiss if I didn't express my gratitude to my lovely wife Arunima. Aru, your love, support and strength made this thesis possible.

Contents

List of Tables

List of Figures

Dedicated to my mother, Susmita Ghosh

Chapter 1

Introduction

Computer vision systems aim to infer properties of the world from rich visual information provided by images and videos. Several systems for automatically detecting and recognizing objects, tracking their motion through image sequences and recovering the 3D world occupied by them have been developed. These systems find use in diverse applications such as augmented reality, multimedia retrieval, robot perception and navigation, remote sensing and biological cell sorting and counting. With acquisition of images, videos and related modalities such as depth, getting progressively cheaper, the impact of systems and techniques developed for computer vision problems will likely grow in the coming years.

Snapshots of the world captured in images and videos arise from inherently lossy projections of 3D scenes onto 2D spaces. As a result, reasoning about the scene depicted in an image is an under constrained problem, and can only be solved under assumptions that constrain the problem. Large variations in shapes, sizes and appearances of objects and regions constituting the real world lead to further complications.

Statistical methods provide an elegant and powerful framework for reasoning under such uncertainty and are widely used for developing solutions robust to difficulties exhibited by visual data. Typically, instead of reasoning about pixels in isolation, statistical methods reason about pixels in context, combining local evidence from pixels with globally consistent interpretations. In this thesis, we extensively utilize the statistical framework and develop new methods and models for two complimentary problems – understanding scenes depicted in images and videos by decomposing them into constituent regions and understanding objects through the discovery and analysis of their parts.

1.1 Motivation

Regions and parts extracted from images, videos and their constituent objects provide an important intermediate representation of visual data. Apart from physiological evidence [1, 2] for part-based representations of objects in the brain, they are useful for several applications in high-level computer vision and beyond. Here, we briefly review a few motivating examples.

1.1.1 Object detection and scene parsing

Recent years have seen significant progress in object recognition and image labeling [3, 4]. State-of-the-art systems place bounding boxes around objects in images and optionally produce dense labelings of images into a predefined set of semantic classes. Such bounding boxes are typically localized by sliding windows at multiple scales and multiple offsets over the entire image. While popular, typical sliding window approaches need to evaluate object detectors at all locations in the image. This is computationally expensive and gets harder to scale with increasing numbers of images and object categories. An alternative direction explored in recent work [5, 6], instead explicitly proposes regions to evaluate object detectors at, leading to more efficient detection. This efficiency allows for the use of computationally expensive features leading to significant performance boosts [5]. In [6], region proposals allow the authors to perform precise object localization, in spite of using a large receptive field convolution neural network. These recent advances demonstrate the promise of improving object detection and localization through improved region discovery from images.

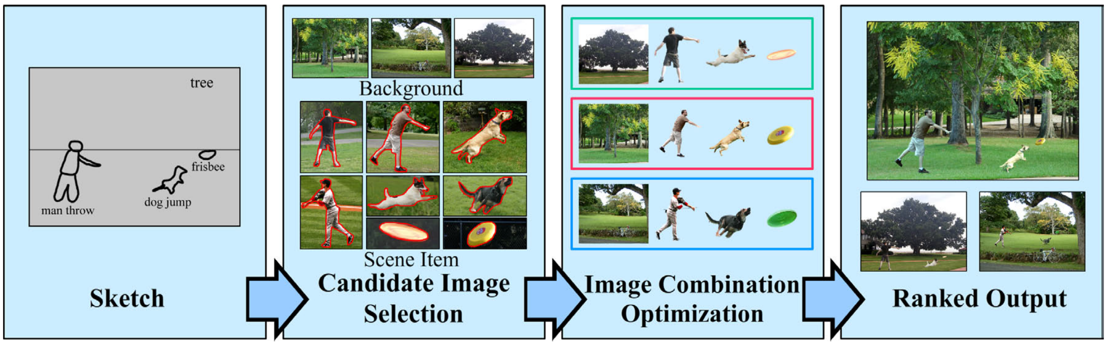


FIGURE 1.1: The sketch2photo system as described in [7]. ©ACM 2009.

1.1.2 Image and video retrieval

Image retrieval was an early motivation for developing effective segmentation algorithms [8]. A more interesting recent case study in this domain is provided by the sketch2photo system [7]. Given a human sketch annotated with a few high level labels the system generates semantically similar images (Figure (1.1)). A vital component of this pipeline is a segmentation routine that extracts salient regions from images retrieved by a retrieval system. The extracted regions are then recombined to generate images semantically close to the human provided sketch. However, the authors find that general purpose segmentation algorithms perform too unreliably to be incorporated into their pipeline. Instead they resort to manually filtering out images with complicated backgrounds and using saliency based foreground segmentation schemes. Better algorithms for discovering regions from images and videos, would clearly benefit such systems.

1.1.3 Metamorphosis

Object animation systems often depend on decomposing a 3D representation of an object into constituent parts. For example, consider the task of metamorphosis [9], where a 3D mesh is transformed into another through a sequence of meshes (Figure (1.2)). Complex objects with several parts need to allow part specific deformations for realistic effects. This necessitates a decomposition of the mesh into underlying parts. In this thesis, we will develop techniques for reliably discovering parts from articulated meshes that can improve metamorphosis pipelines among other applications.

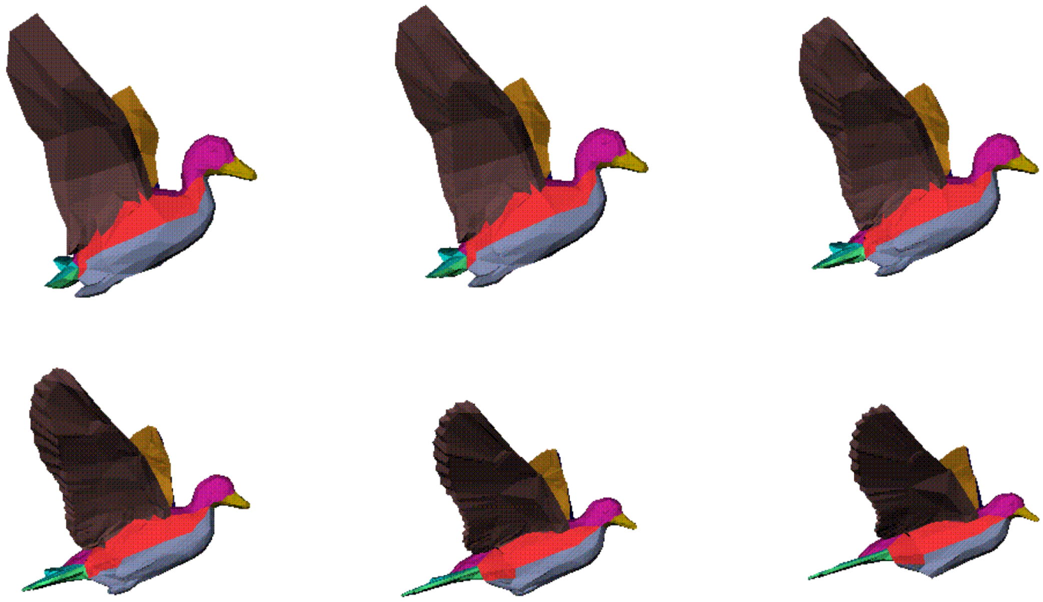


FIGURE 1.2: Metamorphosis of a duck into a dove [9]. ©Eurographics 2002.

1.2 Challenges - It is a complicated world

The world is full of complexity. Objects and their collections that make up the physical environment around us, exhibit a wide range of textures, shapes and sizes. The interaction of light with the material properties of the physical environment, engenders a diverse gamut of colors. Visual snapshots of the world reflect this diversity and in turn vary widely in complexity. Distortions, viewpoint and occlusion effects introduced by the imaging process further confounds the situation. Consequently, image and video collections display wide variability in the number, appearances and sizes of objects and regions they depict. Further, humans interpret and reconstruct the underlying 3D world from images and videos by reasoning at different levels of granularity, abstracting away or focusing on the details as necessary. For example, Figure (1.3) displays images of varying complexity, along with corresponding human interpretations. Different annotators reason about the same image at different levels of detail.

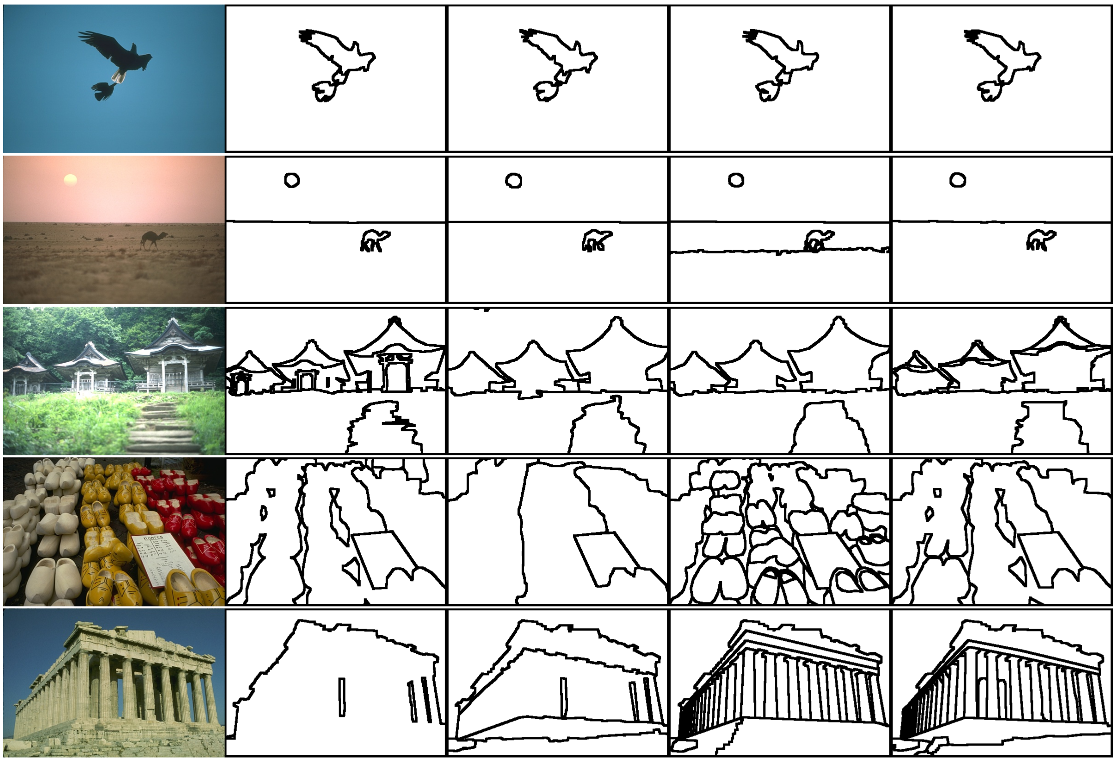


Figure 1.3: A subset of images from the Berkeley image segmentation dataset [10]. Images as well as their human interpretations vary widely in complexity. In this thesis, we develop statistical models that explicitly capture this variability and provide multiple plausible segmentation hypotheses.

Statistical models attempting to extract semantic structure from visual data must robustly cope with uncertainty in the number, shape, size, scale, spatial extent and appearance of scenes and their constituents. In this thesis, we develop Bayesian nonparametric (BNP) statistical approaches for reasoning under such uncertainty. Our models advance recent BNP literature to better reflect the statistics of images, videos and structured data, more broadly. We also develop reliable and effective inference algorithms for exploring the multi-modal posteriors induced by the sophisticated BNP models.

1.3 Thesis Organization

This thesis is organized into the following chapters.

1.3.1 Background

In Chapter 2 we survey the current state-of-the-art and introduce basic building blocks used in subsequent chapters.

1.3.2 Layered Image Segmentation

We focus on segmenting monocular natural images into a set of depth ordered layers in Chapter 3.1. We infer the cardinality of the set automatically, conditioned on the image. Building on the work of [11] we model image partitions through a collection of thresholded functions sampled from Gaussian processes. We then develop novel learning and variational inference algorithms which allow efficient, robust and reliable recovery of layers from natural images. We find that the recovered image partitions are competitive with state-of-the-art image segmentation techniques on standard benchmarks.

1.3.3 Articulated Object Segmentation

In Chapter 4, we consider the problem of articulated 3D object segmentation. We develop a statistical model that combines a prior over object partitions with expressive likelihood distributions over affine transformations. Our model is able to learn both the number and extent of independently deforming object parts from unlabeled data. We demonstrate state-of-the-art performance on a collection of human 3D scans of widely varying shapes and in widely varying poses.

1.3.4 Distributions over Hierarchical Partitions

We develop hierarchical models necessary for modeling partitions over multiple related groups of data in Chapter 5. Approximate inference in these hierarchical models is difficult, requiring the development of novel MCMC algorithms for exploring the intractable posteriors. We apply our models and algorithms to the tasks of activity discovery from MoCap sequences and discourse discovery from textual data to demonstrate the flexibility of the developed models.

1.3.5 Learning Distributions over Partitions

In Chapter 6, we develop algorithms for learning the distance dependent models discussed in Chapter 4 and Chapter 5, from human labelled clusterings. Leaning on recent advances in approximate Bayesian computation (ABC), we develop task loss aware model calibration algorithms that lead to significant performance improvements over hand-crafted models.

1.3.6 Contributions and Recommendations

In Chapter 7, we conclude the thesis by summarizing our contributions and recommending promising avenues of future research.

Chapter 2

Background

This chapter provides an overview of the problems and methods discussed in this thesis. We begin by introducing the problems considered in this thesis and proceed to discuss the current state-of-the-art and their limitations. Next, we introduce basic building blocks utilized by models and algorithms developed in subsequent chapters.

2.1 Image Segmentation

Image segmentation is the problem of partitioning an image into self-similar groups of spatially adjacent pixels. Segmentations provide an important mid-level representation which can be leveraged by various vision tasks including object detection and recognition [12, 13], tracking [14], motion [15] and shape estimation, as well as content based image retrieval [8]. Unsurprisingly, image segmentation has been an active area of research and has produced a large body of research. Here, we briefly review popular image segmentation algorithms and direct the interested reader to [16] for a more comprehensive survey.

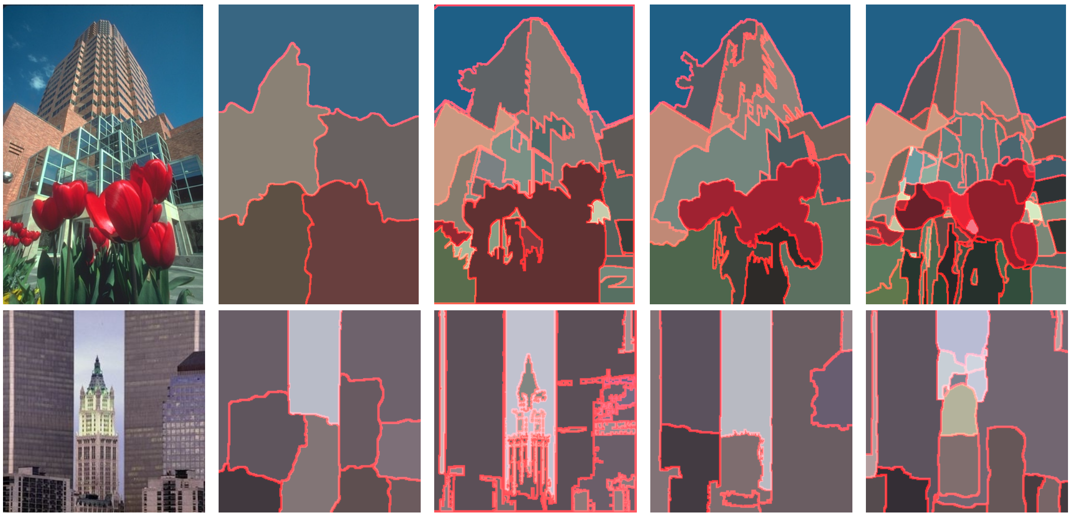


Figure 2.1: Typical results produced by popular image segmentation algorithms. From left to right: Normalized cuts [17], Graph based segmentation of [18], Mean shift [19] and gPb [20].

2.1.1 Methods and Models

Graph partitioning approaches have been widely used for image segmentation. Images are set up as graphs whose vertices correspond to pixels and edges represent dissimilarities between pixels. An "optimal" cut partitioning this graph then produces a segmentation of the underlying image.

Early graph partitioning approaches [21] focused on the well studied problem of finding a minimum cut of a graph. However, the minimum cut criterion isn't well suited for segmentation and leads to partitions with small isolated segments. The approach of [17], an influential piece of work in this area, instead introduces a normalized cut criterion that favors partitions which discourage singleton (and small) components. Drawing on results from spectral graph theory, the authors develop an efficient algorithm for finding normalized cuts of a graph by solving a generalized eigenvalue problem. When applied to natural images the algorithm prefers segmentations containing several approximately uniform sized segments (Figure (2.1)), a byproduct of the normalized cut criterion. In contrast, sizes [22] and boundary lengths [23] of human produced segments exhibit power-law behavior. As a result, the algorithm is rarely used for extracting "human like" segments from natural images. Instead, it is frequently employed for dividing images into large collections of small, roughly equal sized segments (superpixels), a useful preprocessing step that reduces computational burden in many applications. Greedy algorithms have also been explored for graph based image segmentation. For example, in [18] the authors present an agglomerative algorithm that attempts to balance intra-segment dissimilarity with inter segment uniformity. The greedy nature of the algorithm makes it computationally inexpensive compared to the global spectral clustering algorithms.

An orthogonal research direction has focussed on reliably extracting contours from images. Seminal work in this area can be traced back to Roberts [24] and Prewitt [25], who convolved simple gradient filters with an image to generate edge responses. Such gradient responses are often too noisy to be directly useful. More recent work [26] has focused on learning the mapping between edge responses and image cues from collections of natural images. Here, the authors use a logistic regression model for modeling the probability of the presence of a contour conditioned on cues capturing intensity, color and texture gradients. They show that such learned detectors outperform hand crafted gradient filters.

Contour detectors typically do not guarantee closed contours, thus contour detections do not directly translate to segmentations. As a result, contour completion is an active area of research. Approaches for contour completion range from procedures [27–29] for chaining together strong edge fragments to statistical models [30, 31] that reason about contours globally. A popular and particularly simple approach [20, 32] for extracting segmentations from contours involves applying a watershed transformation (or a minor variant) on the contour detector response to produce an over-segmentation. This is generally followed by a greedy agglomerative merging algorithm to produce a nested tree of segmentations. In [20, 33] the authors show that such a procedure combined with a powerful contour detector produces state-of-the-art results.

Methods and models for density estimation of pixel feature spaces have also been proposed for image segmentation. Popular models include those based on finite mixtures [8, 34, 35] and Markov random fields [36]. In “blob world” [8] the authors represent pixels using feature vectors consisting of color, texture responses and pixel locations. These features are then modeled using a Gaussian mixture model.

The model assumes that the appearance and spatial extent of image segments can both be described using Gaussian distributions. While this leads to efficient computational algorithms, in practice, such Gaussianity assumptions prove to be too restrictive. In addition to the overly simplistic unimodal appearance assumption, endowing pixel locations with Gaussian distributions biases segment shapes towards ellipses and produces unwanted segmentation artifacts. Recent work attempts to alleviate these concerns through less restrictive assumptions on segment shapes and appearances. In [22, 37] statistical models that condition on pixel locations instead of generating them have been developed. These models encourage spatially smooth allocation of pixels to segments without placing restrictions on segment shapes. The unimodal appearance assumptions are relaxed in [34] via mixtures of kernel density estimators and in [38] through mixtures of Gaussians (instead of Gaussians). Others [19] have focused on extracting modes of the pixel feature density. In [19] the authors combine a mode seeking algorithm based on the Mean Shift procedure [39] with a nonparametric kernel density estimator. The kernel density estimator estimates the empirical density of the feature space whose modes are estimated by the mean shift algorithm. The local modes are then clustered to eliminate near-overlapping modes. The clustered modes provide the desired image segmentation.

There also exists a large body of work on interactive image segmentation [37, 38, 40–42] where image partitions are modeled using random fields. These approaches require varying degrees of user supervision and typically use combinatorial optimization approaches [43] to infer most likely segmentations.

2.1.2 Benchmarks and Metrics

Perhaps the most popular image segmentation benchmark is the Berkeley segmentation dataset (BSDS) [10]. The benchmark provides several different human annotations for each image, thus providing an empirical measure of annotator variability. It also provides a standard train/test split. However, the diversity of the images in the benchmark is somewhat limited. It predominantly consists of high quality images of outdoor scenes and people captured by skilled photographers. Labelme [44] is a benchmark with complimentary strengths. It is a significantly larger dataset consisting of images from widely varying scenes. However, images here contain only a single ground-truth segmentation which may have been crowd-sourced by several human annotators.

2.1.2.1 Metrics

Quantifying results of unsupervised segmentation is non trivial. Instead of relying on a single metric, researchers often report a number of metrics with complimentary strengths. We briefly review some popular metrics below.

Rand Index Originally proposed in [45], Rand index provides a measure of similarity between two partitions of a set. It computes the ratio of the number of pairwise agreements among elements of the sets with all possible pairwise relationships. Consider two partitions S and S' of a set containing N elements. The Rand index is then given by:

$$RI(S, S') = \frac{1}{\binom{N}{2}} \sum_{i<j} (\mathbf{1}_{i=j} + \mathbf{1}_{i \neq j}), \tag{2.1}$$

Here, $\mathbf{1}_{i=j}$ is an indicator function that is 1 only when elements i and j are both members of the same partition component in S as well as S', $\mathbf{1}_{i \neq j}$ is 1 when i, j are members of distinct components both in S and S'. If S and S' are identical the pair achieves a Rand index of 1. Rand index tends to zero with increasing disagreements between S and S'.

When comparing a partition against multiple other partitions of a set we average individual Rand indices. The resulting quantity is called the probabilistic Rand index. Although Rand index based measures are widely used, they suffer from a small dynamic range [46] and may obfuscate the distinction between different algorithms.

Variation of Information Variation of information (VI) [47] is a metric motivated by information theory. Here a partition S with K components is represented as a categorical random variable taking one of K values with probability $p_k = N_k/N$. The entropy associated with a partition is then defined as

$H(S) = -\sum_{k=1}^{K} p_k \log(p_k)$, the uncertainty associated with an element of S belonging to any component k.

$$\begin{aligned} VI(S, S') &= H(S) + H(S') - 2I(S, S') \\ &= H(S \mid S') + H(S' \mid S). \end{aligned} \tag{2.2}$$

Intuitively, it measures the uncertainty in a partition having observed the other. Clearly, variation of information goes to zero when S and S' are identical. In [47] VI is shown to be upper bounded by $\log(N)$.

Segmentation Covering In [20] the authors measure similarity between two segmentations using a segmentation covering metric,

$$C(S', S) = \frac{1}{N} \sum_{r \in S} |r| \max_{r' \in S'} O(r, r'), \tag{2.3}$$

where $O(r, r') = \frac{|r \cap r'|}{|r \cup r'|}$. The segmentation cover measure usually has a larger dynamic range than Rand index and sometimes allows for better discrimination amongst competing segmentation algorithms.

2.1.3 Limitations of existing approaches

Despite the large amount of research devoted to segmentation, existing state-of-the-art approaches exhibit various limitations.

One challenge is to move beyond seeking a single "optimal" image partition, and to recognize that while there are commonalities among multiple human segmentations of the same image, there is also substantial variability [10]. Most existing segmentation algorithms are endowed with a collection of tunable parameters; a particular configuration may work well on some images, and poorly on others. Often these parameters are tuned via manual experimentation, or expensive validation experiments. Noting this issue, Russell et al. [48] produced a "soup of segments" by varying the parameters of the normalized cuts algorithm, and collecting the range of observed outputs. Others have used agglomerative clustering methods to produce a nested tree of segmentations [20]. A limitation of these procedures is that they fail to provide image-specific estimates of which particular segmentations are most accurate. More generally, segmentation procedures that lack clear probabilistic interpretations have difficulty quantifying uncertainty in the produced segmentations.

Existing probabilistic models do account for uncertainty. However, models based on finite mixtures [8, 34, 35] make overly restrictive assumptions about segment shapes and appearances [8], and require the number of segments to be pre specified [34, 35]. Markov random field based unsupervised image segmentation approaches [36] induce prior distributions that place low probability on human produced image segmentations [49].

Recent advances in Bayesian nonparametric statistical approaches provide a promising direction for alleviating model selection issues. These models [22, 50–52] reason about prior and posterior distributions on the space of image partitions thus considering segmentations of all possible resolutions. In contrast with parametric segmentation models based on finite mixtures or Markov random fields they do *not* require the number of segments. Inference algorithms developed for these models automatically provide calibrated estimates of the relative probabilities of segmentations with varying numbers of regions. Further, the work presented in [22] develops priors over image partitions that closely match statistics of segmentations produced by humans.

In spite of their promise the adoption of these sophisticated models has lagged owing to difficulties in learning and performing inference with these models. In subsequent chapters of this thesis, we address these issues by developing efficient, effective and reliable inference algorithms.

2.2 Video Segmentation

Video segmentation like image segmentation seeks self similar groups of pixels from image sequences. The segments are expected to exhibit appearance homogeneity, spatial consistency and temporally coherent motion.

Although dwarfed by image segmentation, a substantial body of work on video segmentation exists. Several video segmentation procedures can be viewed as temporal extensions of existing image segmentation algorithms. In [53], the authors extend the graph based image segmentation algorithm in [18] by defining spatio-temporal graphs where pixels connect to spatial as well as temporal neighbors. The resulting segmentation is further refined using an agglomerative region merging procedure. The mean shift [19] algorithm for images is extended to videos in [54] and in [55] "blob world" [8] is extended to videos. With such extensions, many of the original image segmentation limitations carry over to videos. Others [56, 57] have focused on tracking of foreground regions. Although, these methods only generate a binary segmentation, they are able to track regions across long video sequences.

Motion Segmentation An interesting direction of research has focused on identifying distinctly moving regions in videos. Trajectory based motion segmentation [58–60] methods attempt to reliably extract and cluster trajectories of sparse interest points. The resulting clusters of trajectories do not provide dense segmentations which can be optionally derived via further post-processing [58]. Given the two step nature of these algorithms, their success depends critically on the quality of the extracted trajectories. They are unable to recover from errors in trajectory computation. An alternate direction [61–64] has focused on building statistical models for layered decomposition of videos. Layer parts (sprites) are associated with distinct appearance and motion models. Probabilistic inference then leads to simultaneous recovery of both motion and layers (segments). Recent work [65] has shown that such joint estimation of motion and segmentation improves performance on both tasks. While exciting, these approaches tend to be computationally expensive. Scaling such models to large videos is an active area of research. Another drawback stems from having to pre-specify the number of segments; this can be difficult for long video sequences.

2.2.1 Benchmarks and Metrics

Several benchmarks have recently been proposed [56, 57, 66, 67] to facilitate more quantitative comparisons among increasingly large number of video segmentation algorithms. One of the more comprehensive benchmarks (VSB100) was introduced in [67]. It contains 100 videos each annotated by multiple annotators and according to multiple criteria (motion, appearance). It also provides a predefined train (40) and test (60) split.

2.2.1.1 Metrics

The metrics introduced in Section 2.1.2.1 can all be used to quantify video segmentation performance. However, applying them to video frames independently and averaging the results is of limited value. Such a procedure doesn't penalize temporal inconsistencies. Instead, the correct way of benchmarking video segmentation results is to treat the entire sequence as a single partition. Numbers computed on such spatio-temporal blocks do penalizing spatially coherent but temporally inconsistent segmentations. Finally, in [67], the authors introduce a volumetric precision and recall (VPR) metric explicitly for quantifying video segmentation performance. They demonstrate several properties that make it well suited for video segmentation.

2.3 Mesh Segmentation

Mesh segmentation has been widely studied as a static clustering problem, where a single mesh is segmented into "semantic" parts using low-level geometric cues such as distance and curvature [68, 69]. While supervised training data can sometimes lead to improved results [70], there are many applications where such data is unavailable, and the proper way to partition a single mesh is inherently ambiguous. A more comprehensive survey of static mesh segmentation methods can be found in [71].

An alternate direction has focused on searching for parts which deform consistently across many meshes. This is a better-posed problem whose solution is directly useful for modeling objects in motion. Limited previous work has sought to segment a mesh into parts based on observed articulations [72–75]. Here, we briefly summarize this literature. A two-stage procedure is presented in [74]. The authors first minimize a variational functional regularized to favor piecewise constant transformations which are then clustered into parts. Others have proposed segmentation procedures [73, 75] that lack coherent probabilistic models, and thus have difficulty quantifying uncertainty and determining appropriate segmentation resolutions.

Anguelov et al. [72] define a global probabilistic model, and use the EM algorithm to jointly estimate parts and their transformations. They also explicitly model spatial dependencies among mesh faces through a Markov random field. While a seminal piece of work, their approach suffers from a few drawbacks. First, their model does not specify a distribution over the space of all mesh partitions, instead focusing on partitions with an *a priori* fixed number of parts. Estimating the appropriate number of parts is difficult and they propose various heuristics to estimate the number. Next, physically plausible mesh parts are expected to be spatially connected. The model in [72] is unable to ensure such parts, instead having to rely on post hoc connected components operation to enforce spatial connectivity. Recent work has also considered joint mesh alignment and segmentation [76]. However, this approach suffers from many of the issues noted above: the number of parts must be specified *a priori*, parts may not be contiguous, and their EM inference appears prone to local optima.

In Chapter 4, we will develop models and algorithms for addressing these issues.

2.4 Bayesian Nonparametrics

Bayesian nonparametric (BNP) methods define distributions over infinite dimensional spaces of functions [77], probability measures [78], and combinatorial structures such as partitions [79], trees [80] and matrices [81, 82]. They lead to flexible models whose complexity grows and adapts with new observations, with small datasets inducing simple posteriors and large datasets leading to richer predictions. Detailed review of Bayesian nonparametric methods can be found in [83–86]. Here, we briefly discuss a subset of BNP methods that are used extensively in subsequent chapters.

2.4.1 Gaussian Processes

Gaussian processes (GP) [77] are a class of stochastic processes that specify distributions over functions. A function $f(\mathbf{x})$ is said to be distributed according to a GP with a mean $m(\mathbf{x})$ and covariance function $\mathrm{k}(\mathbf{x}, \mathbf{x}')$, denoted $f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), \mathrm{k}(\mathbf{x}, \mathbf{x}'))$, if any finite realization of the function $f(\mathbf{x}_{1...\mathrm{N}}) = [f(x_1), ..., f(x_\mathrm{N})]^\mathrm{T}$ follows a Gaussian distribution:

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix} \right) \tag{2.4}$$

The mean $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x}])$ and covariance functions $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$ completely characterize the properties of functions drawn from a GP. While any real valued function may be used to specify the mean function, the covariance is restricted to the class of positive semi-definite (PSD) functions. A function is PSD if for any choice of $\mathrm{N} \in \mathbb{N}$ and $\mathbf{x} = \{x_1, \ldots, x_\mathrm{N}\}$ the gram matrix K with elements $K_{ij} = k(x_i, x_j)$ is PSD. There are several covariance functions popular in the literature and new ones can be created by composing valid covariance functions together. We refer the reader to [77] for an in-depth exposition. Figure 2.2 illustrates functions sampled from Gaussian processes with a squared exponential kernel. The squared exponential covariance produces smooth functions with high probability. In Chapter 3, we describe a class of models that exploit this smoothness property to define prior distributions over realistic image partitions.

2.4.1.1 Regression

Gaussian processes, by specifying distributions over function spaces, naturally lend themselves to nonparametric Bayesian regression problems. Consider a collection of N data and observation pairs $(\mathbf{x}, \mathbf{t}) = \{(x_i, t_i)\}_{i=1}^\mathrm{N}$ with $\{t_i\}_{i=1}^\mathrm{N} \in \mathbb{R}^1$. We can model t_i as noise corrupted observations of a latent function sampled from a GP

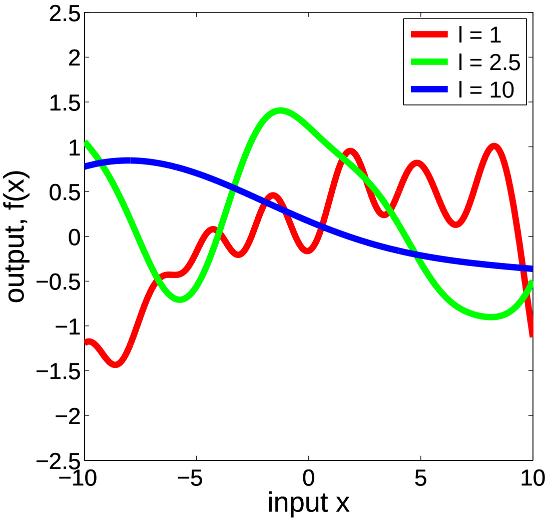


FIGURE 2.2: Functions sampled from a zero mean GP with a squared exponential kernel ($k(x_i, x_j) = exp(-\frac{|x_i - x_j|_2^2}{2\ell^2})$). ℓ is known as the characteristic length scale which governs the expected number of zero crossings exhibited by the function.

with an appropriate mean and covariance function.

$$
\begin{aligned}
f(\mathbf{x}) &\sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \\
t_i &= f(x_i) + \epsilon \\
\epsilon &\sim \mathcal{N}(0, \psi)
\end{aligned}
\tag{2.5}
$$

Using 2.4, we can state the joint distribution specified by the above model:

$$
\begin{aligned}
p(\mathbf{t}, \mathbf{f} \mid \mathbf{x}) &= \mathcal{N}(\mathbf{f} \mid \mathbf{m}, K) \prod_{i=1}^{\mathrm{N}} p(y_i \mid f_i) \\
&= \mathcal{N}(\mathbf{f} \mid \mathbf{m}, K) \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \Psi),
\end{aligned}
\tag{2.6}
$$

where Ψ is a diagonal matrix with $\Psi_{ii} = \psi$ and for notational convenience we have dropped the explicit dependence on x_i and denoted $f(x_i) = f_i$ and $m(x_i) = m_i$.

The marginal likelihood is obtained by marginalizing over the latent function $\mathbf{f}$:

$$
p(\mathbf{t} \mid \mathbf{x}) = \int p(\mathbf{t}, \mathbf{f} \mid \mathbf{x}) d\mathbf{f} = \mathcal{N}(\mathbf{t} \mid \mathbf{m}, K + \Psi) \tag{2.7}
$$

For a set of previously unobserved points $\mathbf{x}_*$ the GP prior specifies the joint distribution over the function evaluations at the new and old locations as follows:

$$p\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}^*\end{bmatrix}\right) \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}\\ \mathbf{m}^*\end{bmatrix}, \begin{bmatrix}K & K_*\\ K_* & K_{*,*}\end{bmatrix}\right) \tag{2.8}$$

Equations 2.7 and 2.8 together give us the joint distribution over $\mathbf{t}$ and the unknown $\mathbf{t}_*$:

$$p\left(\begin{bmatrix}\mathbf{t}\\ \mathbf{t}^*\end{bmatrix} \mid \begin{bmatrix}\mathbf{x}\\ \mathbf{x}^*\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{t}\\ \mathbf{t}^*\end{bmatrix} \mid \begin{bmatrix}\mathbf{m}\\ \mathbf{m}^*\end{bmatrix}, \begin{bmatrix}K+\Psi & K_*\\ K_* & K_{*,*}+\Psi_*\end{bmatrix}\right) \tag{2.9}$$

Using standard Gaussian conditioning ([87]) properties, the posterior can be expressed in closed form as:

$$p(\mathbf{t}^* \mid \mathbf{t}, \mathbf{x}, \mathbf{x}^*) = \mathcal{N}(\mathbf{t}^* \mid \mu_*, \Sigma_*), \tag{2.10}$$

where $\mu_* = K_*^T(K+\psi)^{-1}\mathbf{t}$ and $\Sigma_* = (K_{*,*}+\Psi_*) - K_*(K+\Psi)^{-1}K_*$.

2.4.1.2 Classification

In binary classification, the responses $t_i \in \{+1, -1\}$ are binary random variables. Such binary outputs are typically modeled in the Gaussian process framework as follows,

$$\begin{aligned} f(\mathbf{x}) &\sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \\ p(t_i \mid f_i) &= \Phi(t_i f_i), \end{aligned} \tag{2.11}$$

where $\Phi : R \to [0, 1]$ is a “squashing” function. The nonlinearity introduced by the squashing function produces a non Gaussian, non conjugate posterior over the latent function f. Asa result, approximate inference techniques are necessary to approximate the posterior.

Several techniques [88] for approximating the non Gaussian posterior have been proposed. MCMC methods occupy one end of the spectrum, they can be extremely accurate provided that they are run for a (*often prohibitively*) long duration. Fast deterministic deterministic techniques that approximate the posterior with Gaussian distributions occupy the other end of the spectrum. These are motivated by the observation that when the squashing function is log-concave the posterior turns out to be unimodal [88]. Laplace approximation (LA) is a classic technique in this space. It performs a second order Taylor expansion around the posterior mode to construct a Gaussian approximation to the posterior [77]. Although computationally efficient, the quality of the approximation is often poor, stemming from the fact that the mode can be far from the mean for non Gaussian posteriors. In practice, Expectation propagation (EP) [89] an iterative message passing algorithm is often used. It produces accurate approximations to the posterior while being significantly more efficient than MCMC methods.

In Section 3.3 we will utilize EP to approximate posteriors over image partitions induced by an ordered set of thresholded Gaussian Processes.

2.4.2 Dirichlet Processes

Dirichlet processes (DP) [90] are measures on probability measures, non negative functions which integrate to one. Finite marginals of the Dirichlet process are Dirichlet distributed, just like finite marginals of the Gaussian process follow the Gaussian distribution. Formally, a random distribution $G \sim \mathrm{DP}(\alpha, H)$ is distributed according to a Dirichlet process with a concentration parameter α and a base distribution H over a measurable space Θ, if for any finite measurable partition $\{A_1, \ldots, A_k\}$ of Θ,

$$[G(A_1), \ldots, G(A_k)] \sim \mathrm{Dir}(\alpha H(A_1), \ldots, \alpha H(A_k)). \tag{2.12}$$

The base distribution H acts as the mean of the DP, while the concentration parameter acts as the precision. For any measurable set $A \in \Theta$ we have:

$$\begin{aligned} E[G(A)] &= H(A) \\ Var[G(A)] &= \frac{H(A)(1 - H(A))}{(\alpha + 1)}. \end{aligned} \tag{2.13}$$

The larger the value of α, the more concentrated $G(A)$ is around $H(A)$. Blackwell [91] showed that random distributions drawn from the Dirichlet process are almost surely discrete and place their probability mass on a countably infinite collection of atoms drawn independently from the base distribution H,

$$G = \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k), \tag{2.14}$$

here $\theta_k \sim H$ correspond to the atoms and π_k are the corresponding weights.

2.4.2.1 Stick breaking construction

The implicit characterization of the DP provided in the previous section does not provide a mechanism for sampling distributions from the Dirichlet process. For this, we have to rely on results from Sethuraman [92] who showed that if

$$\begin{aligned} \beta_k &\sim \mathrm{Beta}(1, \alpha) \quad & \theta_k &\sim H \\ \pi_k &= \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \quad & G &= \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k), \end{aligned} \tag{2.15}$$

then $G \sim \mathrm{DP}(\alpha, H)$. When combined with results [90, 91] that prove samples from a DP are discrete and can be represented as in Equation (2.14), Equation (2.15) provides an explicit procedure for sampling from the DP. The mixture weights sampled as in Equation (2.15) are often denoted $\pi \sim \mathrm{GEM}(\alpha)$ in the literature and we will adopt this notation in this thesis. The procedure for constructing the mixture weights $\pi = (\pi_1, \ldots, \pi_\infty)$, may be interpreted as sequentially breaking off pieces from a unit length stick. The first weight is just $\pi_1 \sim \mathrm{Beta}(1, \alpha)$, each subsequent weight π_k is some random fraction (β_k) of the remaining unbroken stick. This analogy engenders the name – "stick breaking" process.

The explicit stick breaking representations play a central role in computations involving Dirichlet processes and prove useful for developing generalizations [22, 93] of the DP.

2.4.2.2 Posterior and Predictive measures

Consider a random measure sampled from a DP, $G \sim \mathrm{DP}(\alpha, H)$ and let $\theta_i \sim G$; $i \in \{1, \ldots, N\}$ denote independent samples from the random measure, then the posterior measure also follows a Dirichlet process [90, 92]:

$$p(G \mid \theta_1, \ldots, \theta_N, \alpha, H) = \mathrm{DP}\left(\alpha + N, \frac{1}{\alpha + N}(\alpha H + \sum_{i=1}^{N} \delta_{\theta_i})\right) \tag{2.16}$$

Further, since a random measure $G \sim \mathrm{DP}(\alpha, H)$ is discrete, $\{\theta_1, \ldots, \theta_N\}$ exhibit a clustering property – there is a strictly positive probability of multiple samples sharing repeated values [90]. Let $\bar{\theta}_1, \ldots \bar{\theta}_K$, $K \leq N$ be the set of unique values exhibited by $\{\theta_1, \ldots, \theta_N\}$.

The realization of a new sample θ_{N+1} is then characterized by a simple predictive rule:

$$\theta_{N+1} | \theta_1, \ldots, \theta_N, \alpha, H \sim \begin{cases} H & \text{w.p. } \frac{\alpha}{\alpha+N}, \\ \bar{\theta}_k & \text{w.p. } \frac{1}{\alpha+N} \sum_{i=1}^{N} \delta(\theta_i, \bar{\theta}_k). \end{cases} \tag{2.17}$$

Observe that by assigning samples θ_i to distinct values $\bar{\theta}_k$, we are implicitly partitioning the data. We can explicitly represent this partition by introducing latent variables z_i that index the set of distinct $\bar{\theta}_k$. Equation (2.17) then immediately leads to the following result,

$$z_{N+1} | z_1, \ldots, z_N, \alpha, H \sim \begin{cases} K + 1 & \text{w.p. } \frac{\alpha}{\alpha+N}, \\ k \in 1, \ldots, K & \text{w.p. } \frac{N_k}{\alpha+N}, \end{cases} \tag{2.18}$$

where $N_k = \sum_{i=1}^{N} \delta(z_i, k)$. This distribution over partitions is popularly referred to as the *Chinese restaurant process* (CRP) [79] and can be described via the following metaphor. Imagine a restaurant with an infinite number of tables. Customers i enter the restaurant in sequence and select a table z_i to join. They pick an occupied table with probability proportional to the number of customers already sitting there, or a new table with probability proportional to the concentration parameter α. The final seating arrangement gives a partition of the data, where each occupied table corresponds to a cluster of the data. From Equation (2.18) it follows that the probability of a partition containing N customers and K occupied components is:

$$p(z_1, \ldots, z_N \mid \alpha) = \frac{\alpha^K \prod_{k=1}^{K} (N_k - 1)!}{\prod_{n=1}^{N} (n - 1 + \alpha)} \tag{2.19}$$

Note that K is a random variable and it can be shown that $E[K] = \alpha \log(N)$ as $N \to \infty$. Dirichlet processes thus prefer models whose complexity grows with data, a characteristic of nonparametric priors. Since Equation (2.19) only depends on the number of occupied components K and its size N_k, we see that although described sequentially, the CRP induces an exchangeable distribution on partitions. The partition probability is invariant to the order in which customer allocations are sampled. Exchangeability plays a key role in deriving MCMC inference algorithms [94] for models that use the CRP representation of the Dirichlet process.

2.4.3 Pitman-Yor Process

The Pitman-Yor [79] process is a generalization of the Dirichlet process. It is specified by the following stick breaking procedure,

$$\begin{aligned} &\beta_k \sim \text{Beta}(1-d, \alpha + kd) \quad \theta_k \sim H \\ &\pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \quad G = \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k), \end{aligned} \tag{2.20}$$

with $0 \leq d < 1$ and $\alpha > -d$. d is known as the discount parameter. Observe that when $d = 0$, we recover the DP. In the general case when $d > 0$, samples from the PY process exhibit heavier tails. To see why, note that the expectation of the random proportion of the stick broken for the k_{th} mixture weight (π_k) is $\mathbb{E}[\beta_k] = \frac{1-d}{1+\alpha+(k-1)d}$, which decreases with increasing k. This suggests that the construction in Equation (2.20) breaks progressively smaller fractions of the unbroken stick, thus preserving more probability mass (length of the unbroken stick) to be distributed amongst subsequent weights. In fact, it can be shown that the number of occupied components under a PY process grows as $O(\alpha N^d)$, in contrast to the logarithmic growth exhibited by the DP. Such a power-law behavior is useful for modeling the size statistics of natural image segments. In Chapter 3 we will utilize the Pitman-Yor process to generate priors over realistic partitions of natural images.

2.4.4 Bayesian Nonparametric Mixtures

Both the Dirichlet and Pitman-Yor processes are distributions over discrete distributions. Such discrete distributions naturally lend themselves to mixture modeling. Consider the following hierarchical model,

$$\begin{gathered} \pi \sim \mathrm{GEM}(\alpha), \\ z_i \sim \pi, \quad x_i \mid z_i \sim F(\theta_{z_i}), \end{gathered} \tag{2.21}$$

where x_i is a random variable sampled from an indexed collection of some parametrized distribution $F(\theta)$ and z_i is a latent variable denoting the unique component (table) responsible for generating x_i. Marginalizing over z we see that the density takes the form of a mixture model with infinitely many components:

$$p(x \mid \pi, \theta_1, \theta_2, \ldots) = \sum_{k=1}^{\infty} \pi_k f(x \mid \theta_k). \tag{2.22}$$

A mixture model with infinite capacity is a powerful tool for data analysis. It allows the complexity of the model to grow with observations (x). The predictive distribution of a new data instance is not limited to the set of components already used to describe existing data. The new data point may be allocated an unoccupied component, if it is sufficiently different from the existing data.

Such models also provide elegant model selection properties. Given a collection of N data instances, the model's posterior distribution has support over the exponentially large set of partitions of the N data points. The modes of the posterior correspond to probable partitions of the data. Inference algorithms that seek such modes then yield both the number of components in the partition as well as the allocation of data instances to components. Throughout this thesis we will make extensive use of this model selection property as we discover parts from *fixed* sized images, videos, 3D meshes and MoCap sequences.

Chapter 3

Spatially Coupled Pitman-Yor Processes for Layered Image Segmentation

This chapter focuses on the problem of extracting depth ordered segments from monocular natural images. We consider a variant of the dependent PY process model introduced in [22]. It captures power law statistics exhibited by human image segments via a stick-breaking construction, and uses an ordered set of Gaussian processes (GPs) to induce spatial dependencies and model occlusion effects. We develop an effective and reliable posterior inference algorithm that is substantially more robust to local optima than previously used algorithms [22]. Our algorithm combines a discrete stochastic search, capable of making large moves in the space of image partitions, with an accurate higher-order variational approximation (based on expectation propagation [89]) to marginalize latent GPs. We improve computational efficiency via a low rank representation of the GP covariance, an innovation that could be applicable to other models with high-dimensional Gaussian variables. Next, we develop algorithms for learning the model hyperparameters, including image-dependent GP covariance functions, from example human segmentations. Together the learning and inference algorithms result in substantial improvements over prior work and demonstrate segmentations that are both qualitatively and quantitatively competitive with state-of-the-art methods.

3.1 Introduction

Image segmentation algorithms partition images into spatially coherent, approximately homogeneous regions. Segmentations provide an important mid-level representation which can be leveraged for various vision tasks including object recognition [12], motion estimation [15], and image retrieval [8]. Despite significant research [17–20, 95], segmentation remains a largely unsolved problem. One major challenge is to move beyond seeking a single "optimal" image partition, and to recognize that while there are commonalities among multiple human segmentations of the same image, there is also substantial variability [10].

Most existing segmentation algorithms are endowed with a host of tunable parameters; a particular configuration may work well on some images, and poorly on others. Often these parameters are tuned via manual experimentation, or expensive validation experiments. Noting this issue, Russell et al. [48] produced a "soup of segments" by varying the parameters of the normalized cuts algorithm, and collecting the range of observed outputs. Others have used agglomerative clustering methods to produce a nested tree of segmentations [20]. A limitation of these approaches is that they do not provide any image-specific estimate of which particular segmentations are most accurate.

In this chapter, we instead pursue a Bayesian nonparametric statistical approach to modeling segmentation uncertainty. We reason about prior and posterior distributions on the space of image partitions, and thus consider segmentations of all possible resolutions. In contrast with parametric segmentation models based on finite mixtures [8, 34, 35] or Markov random fields [36], we do *not* need to prespecify the number of segments. Our inference algorithm automatically provides calibrated estimates of the relative probabilities of segmentations with varying numbers of regions.

Because we define a consistent probabilistic model and not just a segmentation procedure, our approach is a natural building block for more sophisticated models. We improve earlier work on spatially dependent Pitman-Yor (PY) processes [22], which was motivated by the problem of jointly segmenting multiple related images. This PY model was later extended to allow prediction of semantic segment labels, given supervised annotations of objects in training images [96]. Here we focus on the problem of segmenting single images containing unknown object categories.

3.2 Nonparametric Bayesian Segmentation

We have two primary requirements of any segmentation model – a) it should adapt to image complexity and automatically select the appropriate number of segments and b) it should encourage spatial neighbors to cluster together. Furthermore, human segmentations of natural scenes consist of segments of widely varying sizes. It has been observed that histograms over segment areas [10] and contour lengths [23] are well explained by power law distributions. Thus a third requirement is to model this power-law behavior. In this section, we first describe our image representation and then review increasingly sophisticated models which satisfy these requirements. Finally, in Sec. 3.2.4, we propose a novel low-rank model which improves computational efficiency while retaining the above desiderata .

3.2.1 Image Representation

Each image is dicided into roughly 1,000 *superpixels* [97] using the normalized cuts spectral clustering algorithm [17]. The color of each superpixel is described using a histogram of HSV color values with $W_c = 120$ bins. We choose a non-regular quantization to more coarsely group low saturation values. Similarly, the texture of each superpixel is modeled via a local $W_t = 128$ bin texton histogram [98], using quantized band-pass filter responses. Superpixel n is then represented by histograms $x_n = (x_n^t, x_n^c)$ indicating its texture x_n^t and color x_n^c.

3.2.2 Pitman-Yor Mixture Models

Pitman-Yor mixture models extend traditional finite mixture models by defining a Pitman-Yor (PY) process [99] prior over the distribution of mixture components.

The distributions sampled from a PY process are countably infinite discrete distributions which place mass on infinitely many mixture components. Furthermore, these discrete distributions follow a power law distribution and previous work [22] has shown that they model the distribution over human segment sizes well. There are various ways of formally defining the PY process, here we consider the stick breaking representation. Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \ldots)$, $\sum_{k=1}^{\infty} \pi_k = 1$, denote an infinite *partition* of a unit area region (in our case, an image). The Pitman-Yor process defines a prior distribution on this partition via the following *stick-breaking* construction:

$$\begin{aligned} \pi_k &= w_k \prod_{\ell=1}^{k-1} (1 - w_\ell) = w_k \left(1 - \sum_{\ell=1}^{k-1} \pi_\ell\right) \\ w_k &\sim \mathrm{Beta}(1 - \alpha_a, \alpha_b + k\alpha_a) \end{aligned} \tag{3.1}$$

This distribution, denoted by $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_a, \alpha_b)$, is defined by two hyperparameters (the discount and the concentration parameters) satisfying $0 \leq \alpha_a < 1$, $\alpha_b > -\alpha_a$. It can be shown that $\mathbb{E}[\pi_k] \propto k^{-1/\alpha_a}$, thus exhibiting the aforementioned power law distribution.

For image segmentation, each index k is associated with a different segment or region with its own appearance models $\theta_k = (\theta_k^t, \theta_k^c)$ parameterized by multinomial distributions on the W_t texture and W_c color bins, respectively. Each superpixel n then independently selects a region $z_n \sim \mathrm{Mult}(\boldsymbol{\pi})$, and a set of quantized color and texture responses according to

$$p\big(x_n^t, x_n^c \mid z_n, \boldsymbol{\theta}\big) = \mathrm{Mult}\big(x_n^t \mid \theta_{z_n}^t, M_n\big)\, \mathrm{Mult}(x_n^c \mid \theta_{z_n}^c, M_n) \tag{3.2}$$

The multinomial distributions themselves are drawn from a symmetric Dirichlet prior with hyper-paramter ρ. Note that conditioned on the region assignment z_n, the color and texture features for each of the M_n pixels within superpixel n are sampled independently. The appearance feature channels provide weak cues for grouping superpixels into regions. Since, the model doesn't enforce any spatial neighborhood cues, we refer to it as the "bag of features" (BOF) model.

3.2.3 Spatially Dependent PY Mixtures

Next, we review the approach of Sudderth and Jordan [22] which extends the BOF model with spatial grouping cues. The model combines the BOF model with ideas from layered models of image sequences [100], and level set representations for segment boundaries [101].

We begin by elucidating the analogy between PY processes and layered image models. Consider the PY stick-breaking representation of Eq. (3.1). If we sample a random variable z_n such that $z_n \sim \text{Mult}(\boldsymbol{\pi})$ where $\pi_k = w_k \prod_{\ell=1}^{k-1}(1 - w_\ell)$, it immediately follows that $w_k = \mathbb{P}[z_n = k \mid z_n \neq k-1, \ldots, 1]$. The stick-breaking proportion w_k is thus the *conditional* probability of choosing segment k, given that segments with indexes $\ell < k$ have been rejected. If we further interpret the ordered PY segments $\{k = 1, \ldots \infty\}$ as a sequence of layers, z_n can be sampled by proceeding through the layers in order, flipping biased coins (with probabilities w_k) until a layer is chosen. Given this, the probability of assignment to subsequent layers is zero; they are effectively *occluded* by the chosen "foreground" layer.

The spatially dependent Pitman-Yor process of [22] preserves this PY construction, while adding spatial dependence among super-pixels by associating a layer (real valued function) drawn from a zero mean *Gaussian process* (GP) $\mathbf{u}_k \sim GP(\mathbf{0}, \Sigma)$ with each segment k. Σ captures the spatial correlation amongst super-pixels, and without loss of generality we assume that it has a unit diagonal. Each super-pixel can now be associated with a layer following the procedure described in the previous paragraph, n.e.,

$$z_n = \min\left\{k \mid u_{kn} < \Phi^{-1}(w_k)\right\},\ u_{kn} \sim \mathcal{N}(0, \Sigma_{nn} = 1) \tag{3.3}$$

Here, $u_{kn} \perp u_{\ell n}$ for $k \neq \ell$ and $\Phi(u)$ is the standard normal *cumulative distribution function* (CDF). Let $\delta_k = \Phi^{-1}(w_k)$ denote a threshold for layer k. Since $\Phi(u_{kn})$ is uniformly distributed on $[0, 1]$, we have

$$\begin{aligned}
\mathbb{P}(z_n = 1) &= \mathbb{P}(u_{1n} < \delta_1) = \mathbb{P}(\Phi(u_{1n}) < w_1) = w_1 = \pi_1 \\
\mathbb{P}(z_n = 2) &= \mathbb{P}(u_{1n} > \delta_1)\mathbb{P}(u_{2n} < \delta_2) = (1 - w_1)w_2 = \pi_2
\end{aligned} \tag{3.4}$$

and so on. The extent of each layer is determined via the region on which a real-valued function lies below the threshold δ_{layer}, akin to level set methods. If $\Sigma = \mathbf{I}$, we recover the BOF model. More general covariances can be used to encode the prior probability that each feature pair occupies the same segment; developing methods for learning these probabilities is a major contribution of this chapter.

The power law prior on segment sizes is retained by transforming priors on stick proportions $w_k \sim \text{Beta}(1 - \alpha_a, \alpha_b + k\alpha_a)$ into corresponding randomly distributed thresholds $\delta_k = \Phi^{-1}(w_k)$:

$$p(\delta_k \mid \alpha) = \mathcal{N}(\delta_k \mid 0, 1) \cdot \text{Beta}(\Phi(\delta_k) \mid 1 - \alpha_a, \alpha_b + k\alpha_a) \tag{3.5}$$

Figure 3.1 displays corresponding graphical model. Image features are generated as in the BOF model.

3.2.4 Low-Rank Representation

In the preceding generative model, the layer support functions $\mathbf{u}_k \sim \mathcal{N}(0, \Sigma)$ are samples from a Gaussian distribution over N super-pixels. Inference involving GPs involve inverting Σ which is in general a $O(N^3)$ operation and thus scales poorly with increasing image sizes. To cope, we employ a low-rank representation based on $D \leq N$ dimensions, analogous to factor analysis models. We proceed by defining a Gaussian distributed D dimensional latent variable $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, we then set $\mathbf{u}_k = A\mathbf{v}_k + \epsilon_k$, where A is a N-by-D dimensional factor loading matrix and $\epsilon_k \sim \mathcal{N}(0, \Psi)$, with Ψ being a diagonal matrix. Observe that marginalizing over $\mathbf{v}_k$ results in a model equivalent to the full rank model of the preceding section with $\Sigma = AA^T + \Psi$. The low rank model replaces the $O(N^3)$ operation with an $O(ND^2)$ operation, thus scaling linearly with N[1]. Figure 3.1 displays the corresponding graphical model.

[1] A complete time complexity analysis is available in the supplement.

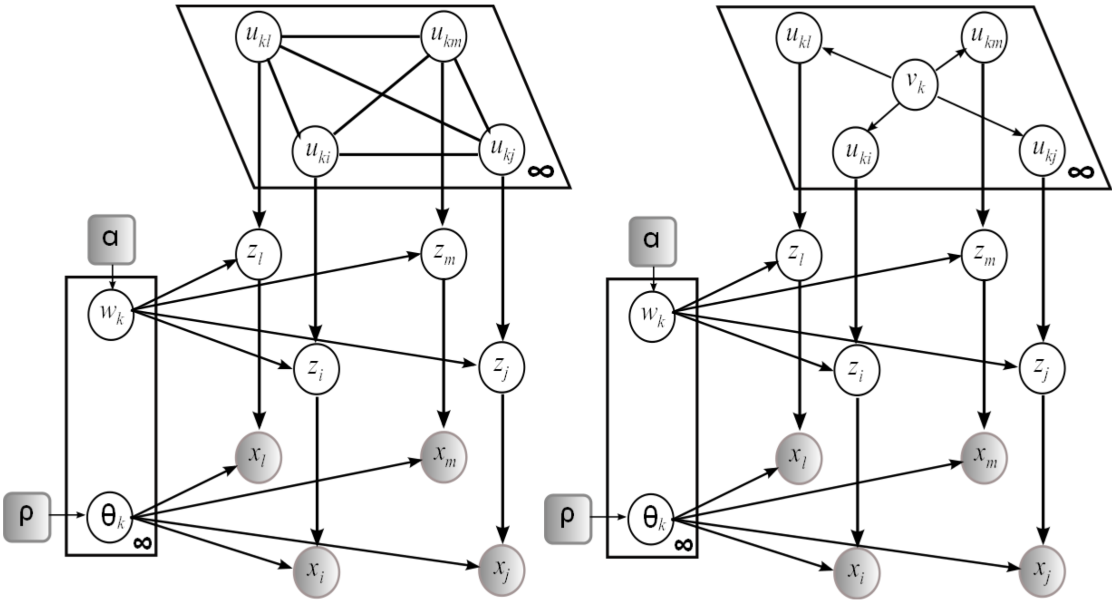


FIGURE 3.1: Generative models of image partitions. *Left.* Spatially dependent PY model, *(right)* low rank model. Shaded nodes represent observed random variables. $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ is a low dimensional Gaussian random variable and $\mathbf{u}_k$ is the corresponding N dimensional layer. $w_k \sim \text{Beta}(1-\alpha_a, \alpha_b + k\alpha_a)$ controls expected layer size and are governed by Pitman-Yor hyper-parameters $\alpha = (\alpha_a, \alpha_b)$. The Dirichlet hyper-parameters $\rho = (\rho^t, \rho^c)$ parametrize appearance distributions. Finally, the color and texture histograms describing super-pixel n are represented as $x_n = (x_n^t, x_n^c)$

3.3 Inference

This section describes a novel, robust to local optima, inference algorithm which is an example of a Maximization Expectation (ME) [102] technique. In contrast to the popular Expectation Maximization algorithms, ME algorithms marginalize model parameters and directly maximize over the latent variables. In our model, the latent variables correspond to segment assignments of super-pixels (z_n). Any configuration of these variables defines a partition of the image. Our strategy is to explore the space of these image partitions by climbing the posterior $p\left(\mathbf{z} \mid \mathbf{x}, \eta\right)$ surface, where $\eta = \{\alpha, \rho, A, \Psi\}$. It is worth noting that since different partitions will have different numbers of segments, we are in fact searching over models of varying complexities akin to traditional model selection techniques.

The algorithm proceeds by first evaluating the posterior for an initial image partition $\mathbf{z}$. It then modifies the partition in an interesting fashion to generate a new partition $\mathbf{z}'$ which is accepted if $p(\mathbf{z}' \mid \mathbf{x}, \eta) \geq p(\mathbf{z} \mid \mathbf{x}, \eta)$. This process is repeated until convergence. By caching the various mutated partitions, we approximate the posterior distribution over partitions (Figure 3.7). In what follows, we first describe the innovations required for evaluating the posterior marginal and then the procedure for mutating a partition.

3.3.1 Posterior Evaluation

In our model (*Figure 3.1*), the posterior $p\ (\mathbf{z} \mid \mathbf{x}, \eta)$ factorizes as $p\ (\mathbf{z} \mid \mathbf{x}, \eta) \propto p\ (\mathbf{x} \mid \mathbf{z}, \rho) p\ (\mathbf{z} \mid \alpha, A, \Psi)$. The likelihood:

$$p\ (\mathbf{x} \mid \mathbf{z}, \rho) = \int_{\Theta} p\ (\mathbf{x} \mid \mathbf{z}, \Theta) p\ (\Theta \mid \rho) d\Theta \tag{3.6}$$

is a standard Dirichlet-multinomial integral and can be evaluated in closed form[2].

Unfortunately, the prior can't similarly be evaluated in closed form. Significant innovations are required for its computation and the remainder of this section details a major contribution of this chapter, an algorithm for evaluating $p\ (\mathbf{z} \mid \eta)$.

$$p(\mathbf{z} \mid \eta) = \prod_{k=1}^{K(\mathbf{z})} \int_{\mathbf{u}_k} \int_{\delta_k} \int_{\mathbf{v}_k} p(\mathbf{z} \mid \delta_k, \mathbf{u}_k) p(\mathbf{u}_k, \mathbf{v}_k \mid A, \Psi)\ p(\delta_k \mid \alpha) d\mathbf{v}_k d\mathbf{u}_k d\delta_k \tag{3.7}$$

where $K(\mathbf{z})$ represents the number of layers in partition $\mathbf{z}$. To simplify notation in the remainder of this chapter we denote $K(\mathbf{z})$ simply by K. Note that in the BOF model $\mathbf{z}$ depends only on α and $p(\mathbf{z}|\alpha)$ can be calculated in closed form:

$$p(\mathbf{z} \mid \alpha) = \alpha_a^K \frac{\Gamma\left(\alpha_b/\alpha_a + K\right) \Gamma(\alpha_b)}{\Gamma(\alpha_b/\alpha_a) \Gamma(N + \alpha_a)} \left(\prod_{k=1}^{K} \frac{\Gamma(M_k - \alpha_a)}{\Gamma(1 - \alpha_a)} \right) \tag{3.8}$$

where N is the number of super-pixels in the partition and M_k is the number of super-pixels in layer k.

Spatial prior evaluation. The integrals in equation 3.7 can be evaluated independently for each layer. In the following analysis, it is implied that we are

[2] The result follows from Dirichlet multinomial conjugacy. Please see the supplement for relevant details

dealing with the k^{th} layer and we drop the explicit dependence on k in our notation. We approximate the joint distribution $p(\mathbf{u}, \mathbf{v}, \delta, z \mid \eta)$ with a Gaussian distribution $q(\mathbf{u}, \mathbf{v}, \delta, \mathbf{z} \mid \eta)$ and the corresponding marginal $p(\mathbf{z} \mid \eta)$ with $q(\mathbf{z} \mid \eta)$, which is easy to compute. We use expectation propagation (EP) [89] to estimate the Gaussian "closest" to the true joint distribution.

Recall that our model assigns super-pixel n to the first layer k whose value is less than the layer's threshold (δ), thus setting $z_n = k$. Equivalently, we can introduce an auxiliary random variable t_n whose value is deterministically related to z_n as follows:

$$t_n = \begin{cases} +1 & \text{if } z_n = k \implies u_n < \delta \\ -1 & \text{if } z_n > k \implies u_n > \delta \end{cases} \tag{3.9}$$

Note that super-pixels with $z_n < k$ have already been assigned to preceding layers and can be marginalized out before inferring the latent Gaussian layer for the k^{th} layer. For a given partition t is known, allowing us to condition on it.

$$\begin{aligned} p(\mathbf{u}, \mathbf{v}, \delta \mid \mathbf{t}, \eta) &= \frac{1}{Z} p(\mathbf{v})\ p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid \mathbf{v}) p(t_n \mid u_n, \delta) \\ p(\mathbf{u}, \mathbf{v}, \delta \mid \mathbf{t}, \eta) &= \frac{1}{Z} \mathcal{N}(\mathbf{v} \mid 0, I)\ p(\delta|\alpha) \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T \mathbf{v}, \psi_n) \mathbb{I}(t_n(\delta - u_n) > 0), \end{aligned} \tag{3.10}$$

where Z is the appropriate normalization constant. Note that the indicator functions $\mathbb{I}(t_n(\delta - u_n) > 0)$ and the threshold prior $p(\delta \mid \alpha)$ are the only non Gaussian terms. We approximate these with un-normalized Gaussians, leading to the following approximate posterior

$$q(\mathbf{u}, \mathbf{v}, \delta \mid \mathbf{t}, \eta) = \frac{1}{Z_{\text{EP}}} \mathcal{N}([\mathbf{u}^T\ \mathbf{v}^T\ \delta]^T \mid \mu_{\approx}, \mathbf{\Sigma}_{\approx}) \tag{3.11}$$

where Z_{EP} ensures appropriate normalization. We now iteratively refine the Gaussian approximation using EP. Applying EP to the low dimensional model requires an interesting combination of Gaussian belief propagation and expectation propagation, the relevant details can be found in the appendix. At convergence, we compute the normalization constant of the approximation

$$Z_{\mathrm{EP}} = p(\mathbf{t} \mid \eta) = \int_{\mathbf{u}} \int_{\mathbf{v}} \int_{\delta} q(\mathbf{u}, \mathbf{v}, \delta, \mathbf{t} \mid \eta), \tag{3.12}$$

the probability of layer k under the spatial prior. Finally, combining Equations 3.6 and 3.12 we have,

$$\log\ p(\mathbf{z} \mid \mathbf{x}, \eta) \propto \gamma\ \log\ p(\mathbf{x} \mid \mathbf{z}, \rho) + \sum_{k=1}^{K} \log Z_{\mathrm{EP}_k}. \tag{3.13}$$

The parameter γ is used to weight the likelihood appropriately. We set $\gamma = \frac{1}{\bar{m}}$, where $\bar{m}$ is the average number of pixels per super-pixel. Recall that our likelihood treats pixels within a super-pixel as independent random variables, necessitating the above down weighting.

3.3.2 Search over partitions

Armed with the ability to evaluate the posterior probability mass for a given image partition, we explore the space of partitions using discrete search. The search performs hill climbing on the posterior surface and explores high probability regions of the partition space. This is similar in spirit to MCMC techniques. Perhaps most similar to our approach is the data driven MCMC approach of Tu *et al.,* [103], which uses a version of the Metropolis-Hastings algorithm along with clever data driven proposals to explore the posterior space. Here, we forgo the requirement of *eventually* converging to the true posterior distribution in exchange for the ease of incorporating flexible search moves and the ability to quickly explore high probability regions of the posterior.

Given a partition we propose a new candidate partition by stochastically choosing one of the following moves:

Merge. Two layers in the current partition are merged into a single layer.

Split. A layer is split into two layers, which are adjacent in layer order. We employ two types of shift moves. Given a layer to be split, the first move works by randomly selecting two seed super-pixels and then assigning all remaining super-pixels to the closest (in appearance space) seed. The initial seeds are chosen such

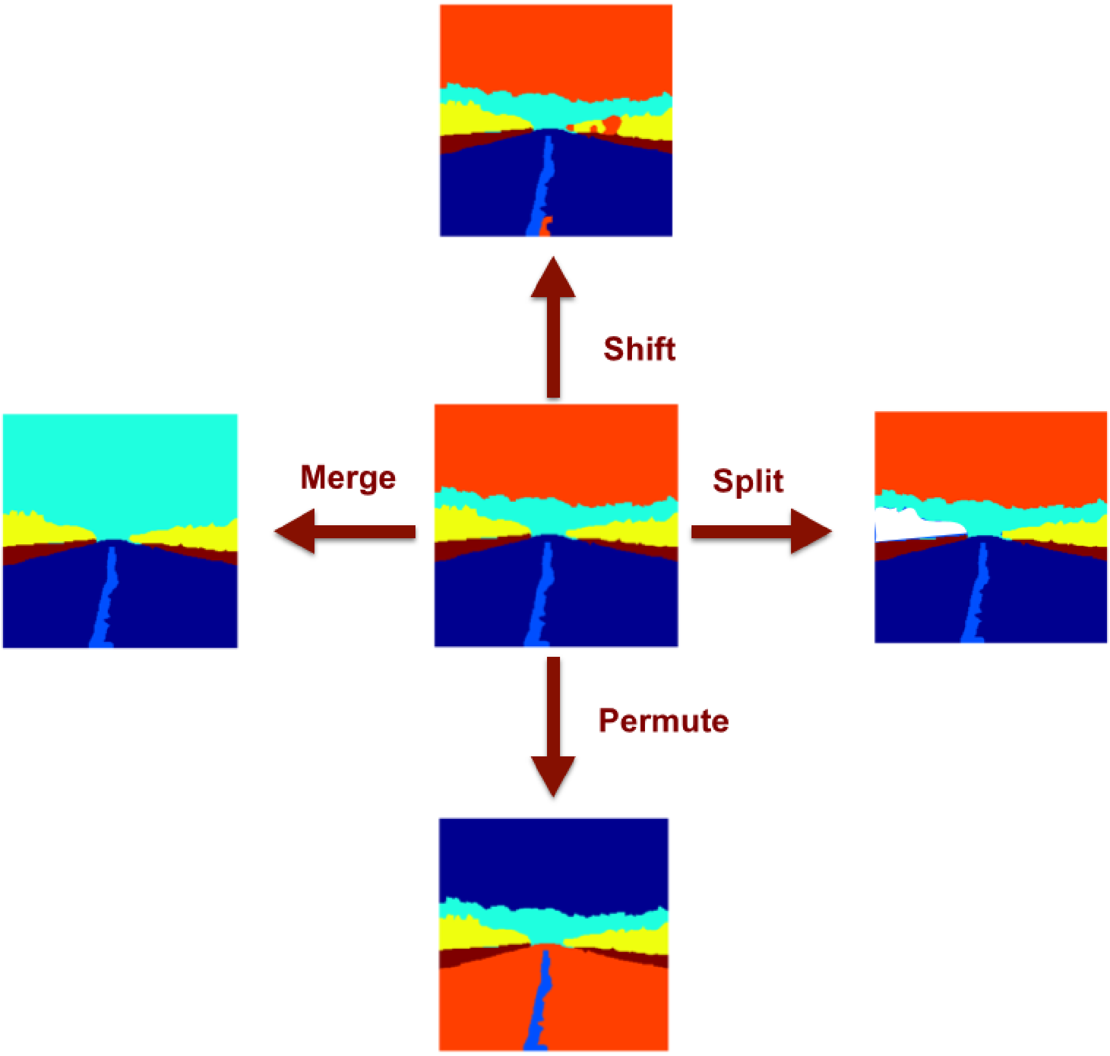


FIGURE 3.2: Illustration of various moves used to explore the space of partitions.

that with high probability they are far in appearance space. The second move employs a connected component operation. If the given layer has disconnected components then one such disconnected component is sampled at random and deemed to be a new layer.

Swap. The swap move reorders the layers in the current partition, by selecting two layers and exchanging their order.

Shift. The shift move refines the partitions found by the other moves. It iterates over all super-pixels in the image assigning each to a segment which maximizes the posterior probability. A naive shift move would evaluate the posterior probability of the partition after every super-pixel shift. This proves to be prohibitively expensive, instead we develop an alternative which allows us to evaluate the posterior after one complete sweep through the super-pixels while ensuring that each individual shift by-and-large increases the posterior [3].

The merge and split moves change the number of layers in a partition performing model selection, while swap and shift attempt to find the optimal partition given a model order.

3.4 Learning from Human Segmentations

In this section, we develop methods for quantitatively calibrating the proposed models to appropriate human segmentation biases. Recall that our model has four hyper-parameters, the PY region size hyper-parameter (α), the appearance hyper-parameter (ρ) and the GP covariance parameters (A and Ψ). We tune these to the human segmentations from the 200 training images of the Berkeley Segmentation Dataset (BSDS) [10]. We show that in spite of the inherent uncertainty in the segmentations of an image, we are able to learn important low level grouping cues.

3.4.1 Learning size and appearance hyper-parameters

The optimal region size hyper-parameters are the ones that best describe the statistics of the training data. We select $\hat{\alpha} = (\hat{\alpha}_a, \hat{\alpha}_b)$ by performing a grid search over 20 evenly spaced α_a and α_b candidates in the intervals $[0, 1]$ and $[0.5, 20]$ respectively and choosing values which maximize the model's likelihood of the training partitions according to equation 3.8. The appearance hyper-parameters $\hat{\rho} = (\hat{\rho}^t, \hat{\rho}^c)$ are tuned through cross validation on a subset of the training set. For BSDS, the estimated parameters equal $\hat{\alpha}_a = 0.15, \hat{\alpha}_b = 1$ $\hat{\rho}^t = 0.01$ and $\hat{\rho}^c = 0.01$

[3] See Appendix for details

3.4.2 Learning covariance kernel hyper-parameters

The covariance kernel governs the type of layers that can be expressed by the model. Estimating it accurately is crucial for accurately partitioning images. In [22, 96] the authors use various heuristics to specify this kernel. Here, we take a more data driven approach and learn the kernel from human segmentations. While we cannot expect our training data to provide examples of all important region appearance patterns, it does provide important cues. Similar to [104], we learn to predict the probability that *pairs* of super-pixels occupy the same segment via human segmentations.

For every pair of super-pixels, we consider several potentially informative low-level cues: (i) pairwise Euclidean distance between super-pixel centers; (ii) intervening contours, quantified as the maximal response of the probability of boundary (Pb) detector [98] on the straight line linking super-pixel centers; (iii) local feature differences, estimated via log empirical likelihood ratios of χ^2 distances between super-pixel color and texture histograms [97]. To model non-linear relationships between these four raw features and super-pixel groupings, each feature is represented via the activation of 20 radial basis functions, with the appropriate bandwidth chosen by cross-validation. Concatenating these gives a feature vector ϕ_{ij} for every super-pixel pair i, j. We then train a L_2 regularized logistic regression model to predict the probability of two super-pixels occupying the same segment q_{ij}. Figure 3.4 illustrates the effect of these cues on partitions preferred by the model.

When probabilities are chosen to depend only on the distance between super-pixels the distribution constructed defines a generative model of image features. When these probabilities also incorporate contour cues, the model becomes a conditionally specified distribution on image partitions, analogous to a conditional random field [105].

From probabilities to correlations. Recall that our layers are functions sampled from multivariate Gaussian distributions, with covariance Σ with unit variance and a potentially different correlation c_{ij} for each super-pixel pair i, j. For each super-pixel pair, q_{ij} is *independently* determined by the corresponding correlation coefficient c_{ij}. Our learning procedure provides estimates of q_{ij}, we now need to map these values to the corresponding correlations c_{ij}.

The mapping between the correlation (c_{ij}) of a pair of Gaussian random variables (u_i and u_j), and the conditionally learned probability q_{ij} of the corresponding super-pixels i and j being assigned to the same layer. According to our model, superpixels i and j are assigned to the same layer k iff both u_i and u_j are less than the threshold δ_k. The probability of this event conditioned on the threshold δ_k is

$$p_{-}(\mathbf{1}_{u_i<\delta_k}\mathbf{1}_{u_j<\delta_k} \mid \delta_k, c) = \int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) du_i du_j \tag{3.14}$$

Marginalizing over the latent thresholds δ_k gives the probability of superpixels i and j being allocated to layer k.

$$\begin{aligned} q_{-}^{k}(\alpha, c) &= \int_{-\infty}^{\delta_k} p^k(\mathbf{1}_{u_i<\delta_k}\mathbf{1}_{u_j<\delta_k} \mid \alpha, c) d\delta_k \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) p(\delta_k|\alpha) du_i du_j d\delta_k \end{aligned} \tag{3.15}$$

The corresponding probability that both u_i and u_j are greater than the δ_k is,

$$\begin{aligned} q_{+}^{k}(\alpha, c) &= \int_{-\infty}^{\delta_k} p^k(\mathbf{1}_{u_i>\delta_k}\mathbf{1}_{u_j>\delta_k} \mid \alpha, c) d\delta_k \\ &= \int_{-\infty}^{\infty}\int_{\delta_k}^{\infty}\int_{\delta_k}^{\infty} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) p(\delta_k|\alpha) du_i du_j d\delta_k \end{aligned} \tag{3.16}$$

Unfortunately neither q_{-} nor q_{+} can be computed in closed form and have to be numerically approximated.

Now observe that superpixels i and j can be assigned to the same layer, if they are both assigned to the first layer or if neither is assigned to the first layer but both are assigned to the second layer or if neither is assigned to the first two layers but both are assigned to the third layer and so on. We can thus express q_{ij} as

$$\begin{aligned} q_{ij} &= q_-^1(\alpha, c) + q_-^2(\alpha, c)q_+^1(\alpha, c) + q_-^3(\alpha, c)q_+^1(\alpha, c)q_+^2(\alpha, c) + \ldots && (3.17) \\ &\approx \sum_{k=1}^{K} q_-^k(\alpha, c) \prod_{l=1}^{K-1} q_+^l(\alpha, c) && (3.18) \end{aligned}$$

where we have explicitly truncated our model to have K (some large number) layers. The above equation defines the sought relationship and allows us to map conditionally learned q_{ij} to pairwise correlations of Gaussian random variables. The mapping is visualized in figure 3.3.

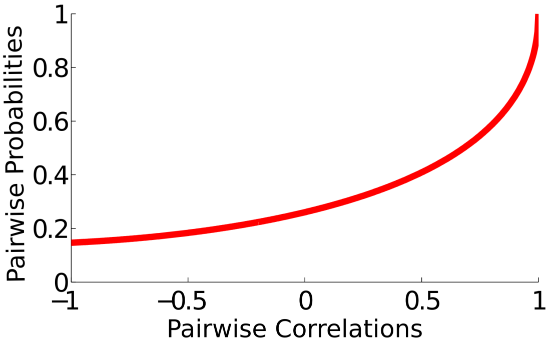


FIGURE 3.3: Mapping between correlation coefficients (c) and pairwise probabilities (q)

The one-to-one mapping between the pairwise probabilities and correlations, allows us to go from logistic regression outputs (q_{ij}) to correlation matrices. These correlation matrices (C), learned from pairwise probabilities will in general not be positive semi-definite (PSD). We cope by finding the closest PSD unit diagonal matrix to the correlation matrix. We use the method of Borsdorf *et al.*, [106], which solves for A and Ψ by minimizing the Frobenius norm$||C - (AA^T + \Psi)||_F$. We note that even the heuristic approaches of Sudderth and Jordan [22] and Shyr textitet al., [96] can yield non PSD correlation matrices. There the authors ensure positive semi-definiteness by performing an eigen-decomposition of C and retaining only non-negative eigenvalues. This is a cruder approximation and leads to poor results.

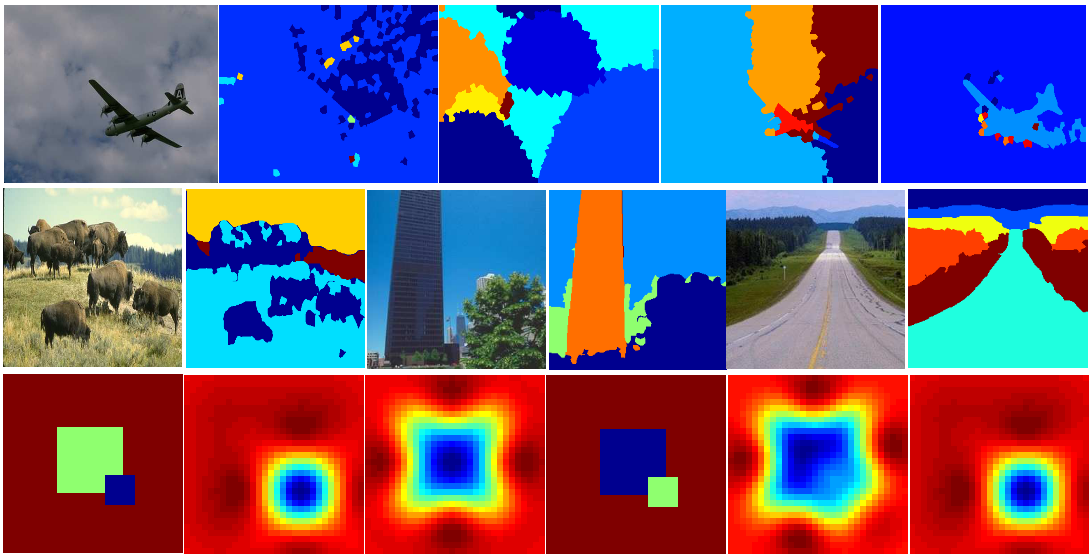


FIGURE 3.4: Model Properties. *TOP*- Prior samples from models employing heuristic distance+pb [22], learned distance (PYdist) , learned distance+pb and all cues (PYall) based covariances. *CENTER*- Layered segmentations produced by our method. *BOTTOM* - Three layer synthetic partitions illustrating preferred layer orderings, Layer 1 is displayed in blue and Layer 2 in green. *Left to right:* Partition 1 (*blue* = *low; red* = *high*), the inferred Gaussian function for layers 1 and 2, partition 2 and the corresponding Gaussian functions. Under our model, partition 1 has a log probability of -77 while partition 2 has a log probability of -90.

3.5 Samples, Layers and Implicit Ordering

In this section, we explore various properties of our model which may not be immediately obvious.

Prior samples The model defines a distribution over image partitions, which can be partially assessed by visualizing partitions sampled from the prior. Figure 3.4 displays such samples. Note that the samples from the conditionally specified models better reflect the structure of the image.

Layers Our model produces partitions made up of layers, not segments. These layers can have multiple connected components, due to either occlusion by a foreground layer, or a layer support function with multimodal shape. The inferred partitions illustrated in the second row of figure 3.4 illustrate this point. The model groups all buffaloes (in the first image), non-contiguous portions of sky, grass and trees (in the second and third images) in the same layer. Traditional segmentation algorithms, having no notion of layers, would assign each non contiguous region to a separate segment. Our layered representation provides a higher level representation of the scene than is possible with a collection of segments, which allows us to naturally deal with complex visual phenomena such as occlusion.

Implicit prior on layer order Recall that a partition is an ordered sequence of layers, and the likelihood of a partition is governed by the likelihood of its constituent layers. Note that reordering layers can change the set of support functions which produce those layers, which in turn makes certain orderings preferable to others. In general, our GP priors prefer simple shapes over complicated ones and hence our model prefers explaining complicated shapes via an occlusion process. Figure 3.4 illustrates these ideas using two synthetic partitions with the order of layers 1 and 2 flipped. The model [4] prefers the partition in the first column over the one in fourth. As can be seen from the inferred layers, partition 1 is explained by the model using simpler Gaussian functions, while partition 2 has to be explained using more complicated and hence less likely Gaussian functions.

3.6 Experiments

In this section we present quantitative evaluations of various aspects of the proposed model along with qualitative results. In all experiments, our model (PYall) used a 200 dimensional low rank representation and ran 200 discrete search iterations, with three random restarts.

Experimental Setup. We benchmark the algorithm on the Berkeley Image Segmentation Dataset (BSDS300 [10]) and a subset of of Oliva and Torralba's [107] eight natural categories dataset. We sampled the first 30 images from each of the eight categories to create a 240 image dataset.

The performance of the algorithms are quantified using the probabilistic Rand Index (*PRI*) [45], and the segmentation covering (*SegCover*) metric [20]. The partitions produced by our model are made up of layers, which may not be spatially contiguous. However, the benchmarks we evaluate on, define segments to be

[4] Here, we have used a squared exponential covariance kernel with length scale set to half of the partition's diagonal length.

spatially contiguous regions. To produce these we run connected components on the layers splitting them into spatially contiguous segments.

Quantifying model enhancements. This chapter improves on both the model (PYheur) and the corresponding inference algorithm presented in [22]. To quantify the performance gains solely from model enhancements we devise the following test. On BSDS300 test images, we compare the log-posterior assigned to the ground truth human segmentations $p(z_{gt}|x, \eta)$ under both models. Since, we already have access to z_{gt} no inference is required and the model which assigns higher probability mass to the ground truth, models the data better. Figure 3.5 presents a scatter plot comparing both models. It is easy to see that PYall models human segmentations significantly better.

Evaluating inference enhancements. Next, we evaluate the performance improvements resulting from the novel inference algorithm[5]. Figure 3.5 displays the result of running mean field and search based inference from 10 random initializations for a given test image. The log-likelihood plots clearly demonstrate mean field being susceptible to local minima. In contrast, EP based search exhibits robustness and all chains converge to high probability partitions. The bottom row displays the best and worst partitions found by mean field and search. As one would expect, there is wide variability in the quality of mean field partitions, while the search partitions are consistently good. The rightmost top row plot displays randomly chosen partitions from the 10 EP search runs. It demonstrates a high correlation between log likelihoods and Rand indices, again verifying that the partitions favored by our model are also favored by humans.

Comparison against competing methods. Our goal is not to produce one "optimal" segmentation but to provide a tractable handle on the posterior distribution over image partitions. Nevertheless, here we demonstrate that by summarizing the posterior with the MAP partition we produce results which are competitive with the state-of-the-art segmentation techniques. We compare against four popular segmentation techniques: Mean Shift (MS) [19], Felzenszwalb and Huttenocher's graph based segmentation (FH) [18], Normalized cuts [17] and gPb contour based

[5] 100 search iterations takes about 30 minutes on a standard quadcore with 4GB of ram.

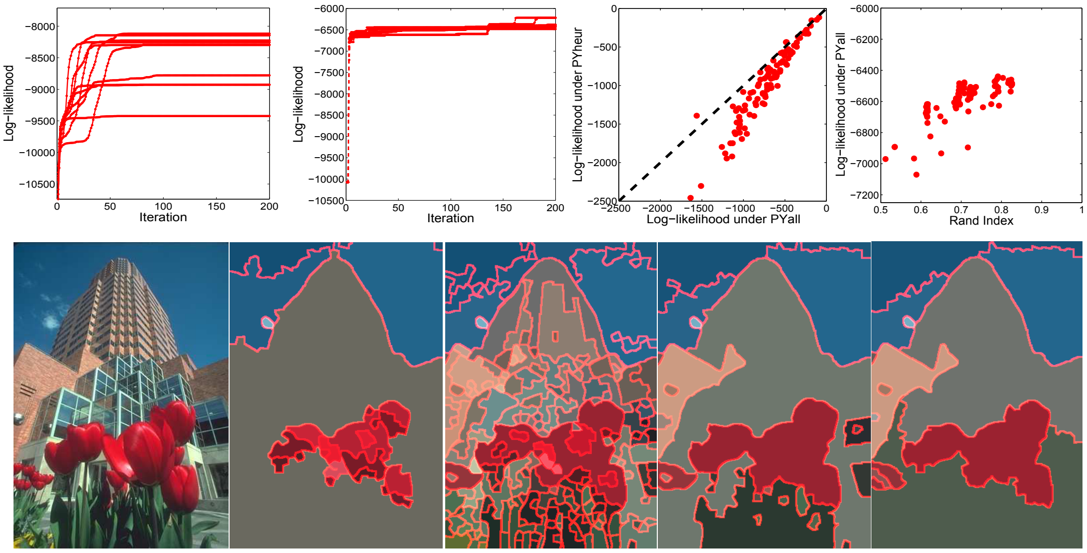


Figure 3.5: *TOP (Left to right)* Log-likelihood (ll) trace plots of mean field runs, search runs, scatter plot comparing PYall and PYheur, scatter plot of ll vs Rand index. *BOTTOM (Left to right)* Test image, partitions with highest and lowest ll found by mean field, best and worst search partitions.

	BSDS300							LabelMe	
	Ncuts	MS	FH	gPb	PYheur	PYdist	PYall	gPb	PYall
PRI	0.73	0.77	0.77	0.80	0.60	0.69	0.76	0.74	0.73
segCover	0.40	0.48	0.53	0.58	0.45	0.50	0.54	0.54	0.55

Table 3.1: Quantitative performance of various algorithms on BSDS300 and LabelMe.

segmentation [20][6]. In addition, we also compare against a version of our model which uses only distance cues for learning the covariance kernel (PYdist). Table 3.1 displays the quantitative numbers achieved on the BSDS300 test set. Figure 3.6 demonstrates qualitative differences amongst the methods. PYall is significantly better than both PYheur and PYdist. According to a Wilcoxon's signed rank test (at an 0.01 significance level) it is also significantly better than Ncuts and MS (on segCover metric, within noise on PRI), within noise of FH and statistically worse than gPb on the BSDS300 dataset.

Next, in order to test generalizability, we compare PYall against the top performing method on BSDS – gPb on the LabelMe dataset. The parameters for either method

[6] All model parameters were tuned by performing a grid search on the training set. See supplement for more details.

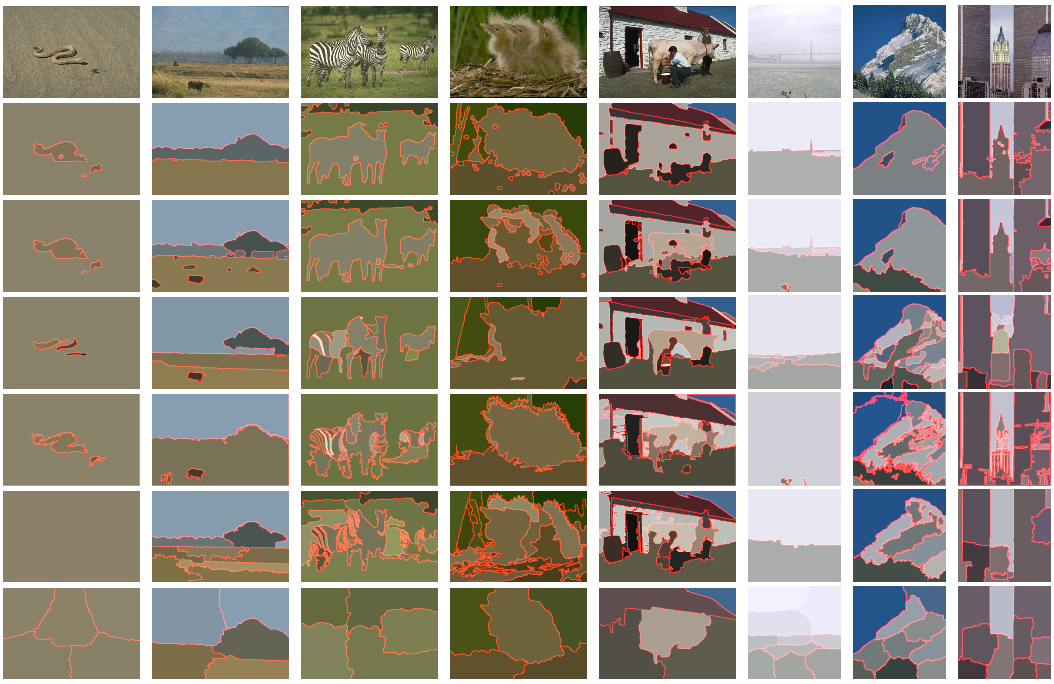


FIGURE 3.6: Comparisons across models. From Top to Bottom: PYdist, PYall, gPb, FH, MS, Ncuts

were tuned on BSDS and were not re-tuned to the LabelMe dataset. Table 3.1 displays the results. PYall and gPb are now statistically indistinguishable.

Posterior Summary. Perhaps, a more accurate assessment of our model involves exploring the posterior distribution over partitions. In Figure 3.7 we summarize the posterior distributions, for a few randomly chosen test images, by presenting a set of high probability partitions discovered by our algorithm. It is worth noting that the set of multiple partitions produced by our method is richer than those produced by a single multi-resolution segmentation tree [20]. For instance, the partitions in the third and fourth columns of the first two rows of Figure 3.7 are mutually inconsistent with any one segmentation tree, but are nonetheless produced by our algorithm. More interesting ways of leveraging the distribution over partitions is an important direction of future work.

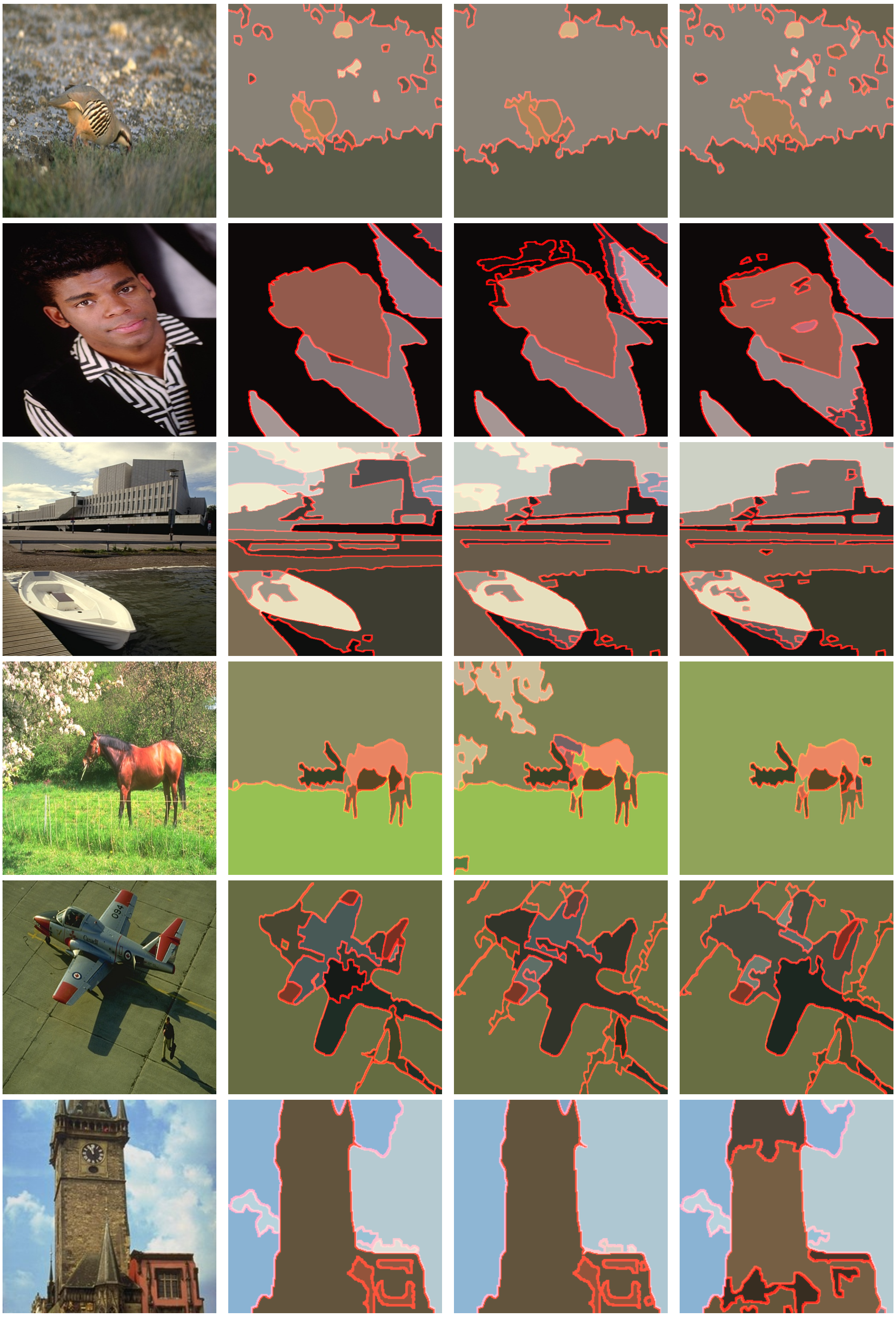


Figure 3.7: Diverse Segmentations. Each row depicts multiple partitions for a given image. Partitions in the second column are the MAP estimates. Other partitions with significant probability masses are shown in the third and fourth columns.

3.7 Discussion

This chapter focused on developing reliable, efficient and effective inference and learning algorithms for dependent PY processes priors over images partitions. We developed substantially improved algorithms for learning from example human segmentations, and robustly inferring multiple plausible segmentations of novel images. The Maximization Expectation algorithm developed here is demonstrably more reliable than corresponding mean field variational inference algorithms, and is broadly applicable for models with Gaussian priors and non-conjugate likelihoods. Together, our learning and inference algorithms provide image segmentation results competitive with the *state-of-the-art*.

Furthermore, by defining a consistent distribution on segmentations of varying resolution, our dependent PY process provides a promising building block for other high-level vision tasks.

Chapter 4

Part Discovery from 3D Objects

The *distance dependent Chinese restaurant process* (ddCRP) [108] is a generalization of the Chinese restaurant process (CRP) Section 2.4.2. Recall that the CRP induces an exchangeable distribution on all possible partitions of a set of objects [79]. While exchangeability provides a computational advantage, from the perspective of approximate inference, it is often an unrealistic assumption when data exhibits strong spatial, sequential or other sorts of dependencies.

The ddCRP relaxes the CRP exchangeability assumption and accommodates random partitions of non-exchangeable data [109]. It alters the CRP by modeling customer links not to tables, but to other customers. The link c_m for customer m is sampled according to the distribution

$$p\left(c_m = n \mid D, f, \alpha\right) \propto \begin{cases} f(d_{mn}) & m \neq n, \\ \alpha & m = n. \end{cases} \tag{4.1}$$

Here, d_{mn} is an externally specified distance between data points m and n, and α determines the probability that a customer links to themselves rather than another customer. D is a matrix of pairwise distances with $D[m, n] = d_{mn}$. The monotonically decreasing decay function $f(d)$ mediates how the distance between two data points affects their probability of connecting to each other. The overall link structure specifies a partition: two customers are clustered together if and only if one can reach the other by traversing the link edges.

The ddCRP can capture a wide variety of correlations among the data through the specification of appropriate choices of distance and decay functions. They have been used for language modeling, clustering time stamped documents and networked data [108]. In this chapter, we use the ddCRP to segment articulated 3D objects. In subsequent chapters, we will develop hierarchical extensions to the ddCRP as well as algorithms for learning distance and decay functions from labeled partitions.

4.1 Distance Dependent Clusters

Like the CRP the ddCRP is a valid distribution over partitions [108] and can be used as an allocation prior in mixture models. In this setting, the ddCRP clusters data in a biased way: each data point is more likely to be clustered with other data that are near it according to the externally specified distance d. This provides us with a flexible class of models that both learn the cardinality of the partition and capture apriori notions of correlations in the data.

The ddCRP mixture generates data as follows:

- For each data instance $i \in [1, N]$ sample a customer link c_i according to equation 4.1. The connected components of the links $c = \{c_i \mid i = 1, \ldots, \mathrm{N}\}$ determine a partition of the dataset $Z(c) = \{z_i \mid i = 1, \ldots, \mathrm{N}\}$.
- For each component $k \in \{1, \ldots, \}$ sample a data generating parameter from a base distribution $\phi_k \sim G_0$.
- Finally, generate data $X = \{x_i \mid i = 1, \ldots, \mathrm{N}\}$ by sampling the data generating distribution $x_i \sim p(x_i \mid \phi_{z_i})$.

Notice that the prior term uses the customer representation to take into account distances between data points while the likelihood term uses the cluster representation to generate observations.

4.1.1 Inference with Gibbs Sampling

The ddCRP mixture has two sets of latent variables the customer assignments c and the cluster specific data generating distribution parameters ϕ_k. The distribution of c conditioned on the observed data X and the model parameters ,with ϕ_k marginalized out is:

$$p(c \,|\, X, \alpha, d, f, G_0) = \frac{\left(\prod_{n=1}^{N} p(c_i \,|\, D, f, \alpha)\right) p(X \mid Z(c), G_0)}{\sum_c \left(\prod_{n=1}^{N} p(c_i \,|\, D, f, \alpha)\right) p(X \mid Z(c), G_0)} \tag{4.2}$$

where $Z(c)$ is the cluster representation derived from the customer representation c.

The posterior in Equation (4.2) is not tractable to compute and we approximate it using Gibbs sampling by iteratively sampling each latent variable c_i conditioned on the others and the observations,

$$p(c_i \,|\, c_{-i}, X, D, \alpha, G_0) \propto p(c_i \,|\, D, \alpha) p(X \,|\, Z(c), G_0). \tag{4.3}$$

The prior term is given in Equation (4.1). We can decompose the likelihood term as follows:

$$\begin{aligned} p(X \mid Z(c), G_0) &= \prod_{k=1}^{K(c)} \int p(\phi_k \mid G_0) \prod_{\{i | Z(c)_i = k\}} p(x_i \mid \phi_k) d\phi_k \\ &= \prod_{k=1}^{K(c)} p(X_{Z(c)=k} \,|\, Z(c), G_0). \end{aligned} \tag{4.4}$$

We have introduced notation to more easily move from the customer representation—the primary latent variables of our model—and the cluster representation. We let $K(c)$ be the number of unique clusters in the customer assignments, $z(c)$ be the cluster assignments derived from the customer assignments, and $X_{Z(c)=k}$ be the collection of observations assigned to the kth cluster. When the base distribution G_0 is conjugate to the data generating distribution the integral in Equation (4.4) is easily computed. In nonconjugate settings, ϕ_k can no longer be analytically marginalized out and an additional layer of sampling is needed to deal with them.

Sampling from Equation (4.3) happens in two stages. First, we remove the customer link c_i from the current configuration. Then, we consider the prior probability of each possible value of c_i and how it changes the likelihood term, by moving from $p(X \mid Z(C_{-i}), G_0)$ to $p(X \mid Z(c), G_0)$.

In the first stage, removing c_i either leaves the cluster structure intact, i.e., $Z(C^{\mathrm{old}}) = Z(C_{-i})$, or splits the cluster assigned to data point i into two. In the second stage, randomly reassigning c_i either leaves the cluster structure intact, i.e., $Z(C_{-i}) = Z(c)$, or joins the cluster assigned to data point i to another. Via these moves, the sampler explores the space of possible segmentations.

Let ℓ and m be the indices of the tables that are joined to index k. We first remove c_i, possibly splitting a cluster. Then we sample from

$$p(c_i \mid c_{-i}, X, D, \alpha, G_0) \propto \begin{cases} p(c_i \mid D, \alpha)\Gamma(X, Z, G_0) & \text{if } c_i \text{ joins } \ell \text{ and } m; \\ p(c_i \mid D, \alpha) & \text{otherwise,} \end{cases} \tag{4.5}$$

where

$$\Gamma(X, Z, G_0) = \frac{p(X_{Z(c)=k} \mid G_0)}{p(X_{Z(c)=\ell} \mid G_0) p(X_{Z(c)=m} \mid G_0)}. \tag{4.6}$$

This defines a Markov chain whose stationary distribution is the posterior of the ddCRP mixture.

4.2 Motion based 3D object segmentation

In this section, we leverage the ddCRP mixture machinery for performing mesh segmentation. Mesh segmentation methods decompose a three-dimensional (3D) mesh, or a collection of aligned meshes, into their constituent parts. This well-studied problem has numerous applications in computational graphics and vision, including texture mapping, skeleton extraction, morphing, and mesh registration and simplification. We focus in particular on the problem of segmenting an articulated object, given aligned 3D meshes capturing various object poses. The meshes we consider are complete surfaces described by a set of triangular faces, and we seek a segmentation into spatially coherent parts whose spatial transformations capture object articulations. Applied to various poses of human bodies

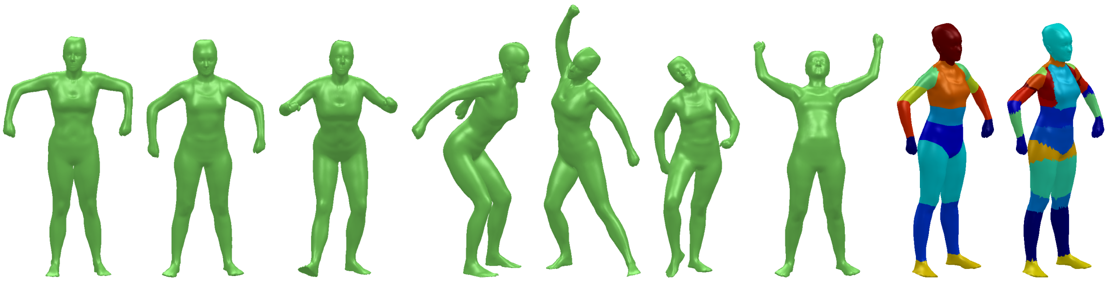


Figure 4.1: Human body segmentation. *Left:* Reference poses for two female bodies, and those bodies captured in five other poses. *Right:* A manual segmentation used to align these meshes [110], and the segmentation inferred by our ddCRP model from 56 poses. The ddCRP segmentation discovers parts whose motion is nearly rigid, and includes small parts such as elbows and knees absent from the manual segmentation.

as in Figure 4.1, our approach identifies regions of the mesh that deform together, and thus provides information which could inform applications such as the design of protective clothing.

Several issues must be addressed to effectively segment collections of articulated meshes. First, the number of parts comprising an articulated object is unknown *a priori*, and must be inferred from the observed deformations. Second, mesh faces exhibit strong spatial correlations, and the inferred parts must be contiguous. This spatial connectivity is needed to discover parts which correspond with physical object structure, and required by target applications such as skeleton extraction. Finally, our primary goal is to understand the structure of human bodies, and humans vary widely in size and shape. People move and deform in different ways depending on age, fitness, body fat, etc. A segmentation of the human body should take into account this range of variability in the population. To our knowledge, no previous methods for segmenting meshes combine information about deformation from multiple bodies to address this *corpus segmentation* problem.

In the rest of this section, we develop a statistical model which addresses all of these issues. We adapt the ddCRP to model spatial dependencies among mesh triangles, and enforce spatial contiguity of the inferred parts [52]. Unlike most previous mesh segmentation methods, our approach allows data-driven inference of an appropriate number of parts, and uses an affine transformation-based likelihood to accommodate object instances of varying shape.

4.2.1 A Part-Based Model for Mesh Deformation

Consider a collection of J meshes, each with N triangles. For some input mesh j, we let $y_{jn} \in \mathbb{R}^3$ denote the 3D location of the center of triangular face n, and $Y_j = [y_{j1}, \ldots, y_{jN}] \in \mathbb{R}^{3 \times N}$ the full mesh configuration. Each mesh j has an associated N-triangle reference mesh, indexed by b_j. We let $x_{bn} \in \mathbb{R}^4$ denote the location of triangle n in reference mesh b, expressed in homogeneous coordinates $(x_{bn}(4) = 1)$. A full reference mesh $X_b = [x_{b1}, \ldots, x_{bN}]$. In our later experiments, Y_j encodes the 3D mesh for a person in pose j, and X_{b_j} is the reference pose for the same individual.

We estimate aligned correspondences between the triangular faces of the input pose meshes Y_j, and the reference meshes X_b, using a recently developed method [110]. This approach robustly handles 3D data capturing varying shapes and poses, and outputs meshes which have equal numbers of faces in one-to-one alignment. Our segmentation model does not depend on the details of this alignment method, and could be applied to data produced by other correspondence algorithms.

4.2.2 Nonparametric Spatial Priors for Mesh Partitions

The ddCRP, endowed with an appropriate distance function, is particularly well suited for modeling segmentations of articulated objects. In addition to allowing data-driven inference of the true number of mostly-rigid parts underlying the observed data and encouraging spatially adjacent triangles to lie in the same part, it *guarantees* that all inferred parts are spatially contiguous.

We define the distance between two triangles as the minimal number of hops, between adjacent faces, required to reach one triangle from the other. A "window" decay function of width 1, $f(d) = \mathbf{1}_{d \leq 1}$, then restricts triangles to link only to immediately adjacent faces. Note that this doesn't limit the size of parts, since all pairs of faces are potentially reachable via a sequence of adjacent links. However, it does guarantee that only spatially contiguous parts have non-zero probability under the prior. This constraint is preserved by our MCMC inference algorithm.

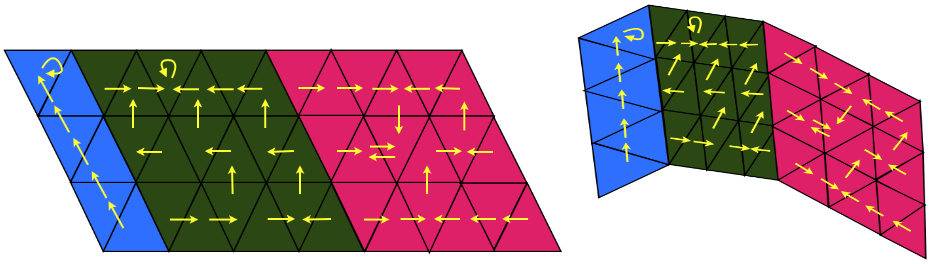


Figure 4.2: *Left:* A reference mesh in which links (yellow arrows) currently define three parts (connected components). *Right:* Each part undergoes a distinct affine transformation, generated as in Equation (4.7).

4.2.3 Modeling Part Deformation via Affine Transformations

Articulated object deformation is naturally described via the spatial transformations of its constituent parts. We expect the triangular faces within a part to deform according to a coherent part-specific transformation, up to independent face-specific noise. The near-rigid motions of interest are reasonably modeled as affine transformations, a family of co-linearity preserving linear transformations. We concisely denote the transformation from a reference triangle to an observed triangle via a matrix $A \in \mathbb{R}^{3\times4}$. The fourth column of A encodes translation of the corresponding reference triangle via homogeneous coordinates x_{bn}, and the other entries encode rotation, scaling, and shearing.

Previous approaches have treated such transformations as parameters to be estimated during inference [72, 76]. Here, we instead define a prior distribution over affine transformations. Our construction allows transformations to be analytically marginalized when learning our part-based segmentation, but retains the flexibility to later estimate transformations if desired. Explicitly modeling transformation uncertainty makes our MCMC inference more robust and rapidly mixing [94], and also allows data-driven determination of an appropriate number of parts.

The matrix of numbers encoding an affine transformation is naturally modeled via multivariate Gaussian distributions. We place a conjugate, matrix normal-inverse-Wishart [111, 112] prior on the affine transformation A and residual noise covariance matrix Σ:

$$\begin{aligned} \Sigma &\sim \mathcal{IW}(n_0, S_0) \\ A \mid \Sigma &\sim \mathcal{MN}(M, \Sigma, K) \end{aligned} \tag{4.7}$$

Here, $n_0 \in \mathbb{R}$ and $S_0 \in \mathbb{R}^{3\times 3}$ control the variance and mean of the Wishart prior on Σ^{-1}. The mean affine transformation is $M \in \mathbb{R}^{3\times 4}$, and $K \in \mathbb{R}^{4\times 4}$ and Σ determine the variance of the prior on A. Applied to mesh data, these parameters have physical interpretations and can be estimated from the data collection process. While such priors are common in Bayesian regression models, our application to the modeling of geometric affine transformations appears novel.

Allocating a different affine transformation for the motion of each part in each pose (Figure 4.2), the overall generative model can be summarized as follows:

1. For each triangle i, sample an associated link $c_i \sim \text{ddCRP}\,(\alpha, f, D)$. The part assignments z are a deterministic function of the sampled links $c = [c_1, \ldots, c_N]$.
2. For each pose j of each part k, sample an affine transformation A_{jk} and residual noise covariance Σ_{jk} from the matrix normal-inverse-Wishart prior of Equation (4.7).
3. Given these pose-specific affine transformations and assignments of mesh faces to parts, independently sample the observed location of each pose triangle relative to its corresponding reference triangle, $y_{ji} \sim \mathcal{N}(A_{jz_i} x_{b_{ji}}, \Sigma_{jz_i})$.

Note that Σ_{jk} governs the degree of non-rigid deformation of part k in pose j. It also indirectly influences the number of inferred parts: a large S_0 makes large Σ_{jk} more probable, which allows more non-rigid deformation and permits models which utilize fewer parts. The overall model is

$$\begin{aligned} p(\mathbf{Y}, c, A, \Sigma \mid \mathbf{x}, b, D, \alpha, f, \eta) = p(c \mid D, f, \alpha) \\ \prod_{j=1}^{J} \left[\prod_{k=1}^{K(c)} p(A_{jk}, \Sigma_{jk} \mid \eta) \right] \left[\prod_{i=1}^{N} \mathcal{N}(y_{ji} \mid A_{jz_i} x_{b_{ji}}, \Sigma_{jz_i}) \right] \end{aligned} \tag{4.8}$$

where $\mathbf{Y} = \{Y_1, \ldots, Y_J\}$, $\mathbf{x} = \{X_1, \ldots, X_B\}$, $b = [b_1, \ldots, b_J]$, the ddCRP links c define assignments z to $K(c)$ parts, and $\eta = \{n_0, S_0, M, K\}$ are likelihood hyperparameters. There is a single reference mesh X_b for each object instance b, and Y_j captures a single deformed pose of X_{b_j}.

4.2.4 Related Work

Mixture of Regression models The mesh-crp and mesh-ddcrp models are instances of the mixture of regressions [113] model. When $J = 1$, the crp-mesh model can be understood as a Bayesian nonparametric mixture of linear regressions [114] while the ddcrp-mesh model is a further generalization accounting for dependencies among covariates. When $J > 1$ the models provide a further generalization – the ability to model multiple outputs.

4.2.5 Inference

We seek the constituent parts of an articulated model, given observed data ($\mathbf{x}$, $\mathbf{Y}$, and b). These parts are characterized by the posterior distribution of the customer links c_i. Following, Section 4.1.1 we develop a collapsed Gibbs sampler, which iteratively draws c_i from the conditional distribution:

$$p(c_i \mid c_{-i}, \mathbf{x}, \mathbf{Y}, b, D, f, \alpha, \eta) \propto p(c_i \mid D, f, \alpha) p(\mathbf{Y} \mid Z(c), \mathbf{x}, b, \eta). \tag{4.9}$$

Here, $Z(c)$ is the clustering into parts defined by the customer links c. The likelihood term in the above equation factorizes as:

$$p(\mathbf{Y} \mid Z(c), \mathbf{x}, b, \eta) = \prod_{k=1}^{K(c)} \prod_{j=1}^{J} p(Y_{jk} \mid X_{b_j k}, \eta) \tag{4.10}$$

where $Y_{jk} \in \mathbb{R}^{3 \times N_k}$ is the set of triangular faces in part k of pose j, and $X_{b_j k}$ are the corresponding reference faces. Exploiting the conjugacy of the normal likelihood to the prior over affine transformations in Equation (4.7), we marginalize the part-specific latent variables A_{jk} and Σ_{jk} to compute the marginal likelihood in closed form (Chapter B):

$$p(Y_{jk} \,|\, X_{b_jk}, \eta) = \frac{|K|^{3/2}|S_0|^{(n_0/2)}\Gamma_3\left(\frac{N_k+n_0}{2}\right)}{\pi^{(3N_k/2)}|S_{xx}|^{(3/2)}|S_0+S_{y|x}|^{((N_k+n_0)/2)}\Gamma_3(\frac{n_0}{2})}, \tag{4.11}$$

$$S_{xx} = X_{b_jk}{X_{b_jk}}^T + K, \;\; S_{yx} = Y_{jk}{X_{b_jk}}^T + MK, \tag{4.12}$$

$$S_{y|x} = Y_{jk}{Y_{jk}}^T + MKM^T - S_{yx}(S_{xx})^{-1}S_{yx}^T. \tag{4.13}$$

Putting Equation (4.5) and Equation (4.13) we have the required posterior distribution:

$$p(c_i \,|\, c_{-i}, \mathbf{x}, \mathbf{Y}, b, D, f, \alpha, \eta) \propto \begin{cases} p(c_i \,|\, D, f, \alpha)\Delta(\mathbf{Y}, \mathbf{x}, b, Z(c), \eta) & \text{if } c_i \text{ links } k_1 \text{ and } k_2; \\ p(c_i \,|\, D, \alpha) & \text{otherwise,} \end{cases}$$

$$\Delta(\mathbf{Y}, \mathbf{x}, b, Z(c), \eta) = \frac{\prod_{j=1}^{J} p(Y_{jk_1 \cup k_2} \,|\, X_{b_jk_1 \cup k_2}, \eta)}{\prod_{j=1}^{J} p(Y_{jk_1} \,|\, X_{b_jk_1}, \eta) \prod_{j=1}^{J} p(Y_{jk_2} \,|\, X_{b_jk_2}, \eta)}. \tag{4.14}$$

Here, k_1 and k_2 are parts in $z(c_{-i})$. Note that if the mesh segmentation c is the only quantity of interest, the analytically marginalized affine transformations A_{jk} need not be directly estimated. However, for some applications the transformations are of direct interest. Given a sampled segmentation, the part-specific parameters for pose j have the following posterior [111]:

$$p(A_{jk}, \Sigma_{jk} \,|\, Y_{jk}, X_{b_jk}, \eta) \propto \mathcal{MN}(A_{jk} \,|\, S_{yx}S_{xx}^{-1}, \Sigma_{jk}, S_{xx})\mathcal{IW}(\Sigma_{jk} \,|\, N_k + n_0, S_{y|x} + S_0) \tag{4.15}$$

Marginalizing the noise covariance matrix, the distribution over transformations is then

$$\begin{aligned} p(A_{jk} \,|\, Y_{jk}, X_{b_jk}, \eta) &= \int \mathcal{MN}(A_{jk} \,|\, S_{yx}S_{xx}^{-1}, \Sigma_{jk}, S_{xx}) IW(\Sigma_{jk} \,|\, N_k + n_0, S_{y|x} + S_0)\, d\Sigma_{jk} \\ &= \mathcal{MT}(A_{jk} \,|\, N_k + n_0, S_{yx}S_{xx}^{-1}, S_{xx}, S_{y|x} + S_0) \end{aligned} \tag{4.16}$$

where $\mathcal{MT}(\cdot)$ is a matrix-t distribution [112] with mean $S_{yx}S_{xx}^{-1}$, and $N_k + n_0 - 2$ degrees of freedom.

4.2.6 Experiments

We now experimentally validate, the *mesh-ddcrp* model developed in the previous sections. Both qualitative and quantitative comparisons are provided. Because "ground truth" parts are unavailable for the real body pose datasets of primary interest, we propose an alternative evaluation metric based on the prediction of held-out object poses, and show that the mesh-ddcrp performs favorably against competing approaches.

We primarily focus on a collection of 56 training meshes, acquired and aligned [110] from 3D scans of two female subjects in 27 and 29 poses. For quantitative tests, we employ 12 meshes of each of six different female subjects [115] (Figure 4.4). For each subject, a mesh in a canonical pose is chosen as the reference mesh (Figure 4.1). These meshes contain about 20,000 faces.

4.2.6.1 Hyperparameter Specification and MCMC Learning

The hyperparameters that regularize our mesh-ddcrp prior have intuitive interpretations, and can be specified based on properties of the mesh data under consideration. As described in Section 4.2.2, the ddCRP distances D and f are set to guarantee spatially connected parts. The self-connection parameter is set to a small value, $\alpha = 10^{-8}$, to encourage creation of larger parts.

The matrix normal-inverse-Wishart prior on affine transformations A_{jk}, and residual noise covariances Σ_{jk}, has hyperparameters $\eta = \{n_0, S_0, M, K\}$. The mean affine transformation M is set to the identity transformation, because on average we expect mesh faces to undergo small deformations. For the noise covariance prior, we set the degrees of freedom $n_0 = 5$, a value which makes the prior variance nearly as large as possible while ensuring that the mean remains finite. The expected part variance S_0 captures the degree of non-rigidity which we expect parts to demonstrate, as well as noise from the mesh alignment process. The correspondence error in our human meshes is approximately 0.01m; allowing for some part non-rigidity, we set $\sigma = 0.015$m and $S_0 = \sigma^2 \times \mathbf{I}_{3\times3}$. K is a precision matrix set to $K = \sigma^2 \times \text{diag}(1, 1, 1, 0.1)$.The Kronecker product of K^{-1} and S_0 governs the covariance of the distribution on A. Our settings make this nearly identity for most components, but the translation components of A have variance which is an order of magnitude larger, so that the expected scale of the translation parameters matches that of the mesh coordinates.

In our experiments, we ran the mesh-ddcrp sampler for 200 iterations from each of five random initializations, and selected the most probable posterior sample. The computational cost of a Gibbs iteration scales linearly with the number of meshes; our unoptimized Matlab[1] implementation required around 10 hours to analyze 56 human meshes.

4.2.6.2 Baseline Segmentation Methods

We compare the mesh-ddcrp model to three competing methods. The first is a modified agglomerative clustering technique [116] which enforces spatial contiguity of the faces within each part. At initialization, each face is deemed to be its own part. Adjacent parts on the mesh are then merged based on the squared error in describing their motion by affine transformations. Only adjacent parts are considered in these merge steps, so that parts remain spatially connected.

Our second baseline is based on a publicly available implementation of spectral clustering methods [117], a popular approach which has been previously used for mesh segmentation [118]. We compare to an affinity matrix specifically designed to cluster faces with similar motions [119]. The affinity between two mesh faces u, v is defined as $C_{uv} = \exp\{-\frac{\sigma_{uv}+\sqrt{m_{uv}}}{S^2}\}$, where $m_{uv} = \frac{1}{J^2}\sum_j \delta_{uvj}$, δ_{uvj} is the Euclidean distance between u and v in pose j, $\sigma_{uv} = \sqrt{\frac{1}{J}\sum_j(\delta_{uvj} - \bar{\delta}_{uv})^2}$ is the corresponding standard deviation, and $S = \frac{1}{M}\sum_{u,v}\sigma_{uv} + \sqrt{m_{uv}}$ for all M pairs of faces u, v.

For the agglomerative and spectral clustering approaches, the number of parts must be externally specified; we experimented with $K = 5, 10, 15, 20, 25, 30$ parts. We also consider a Bayesian nonparametric baseline which replaces the ddCRP prior over mesh partitions with a standard CRP prior. The resulting *mesh-crp* model may estimate the number of parts, but doesn't model mesh structure or enforce part contiguity. The expected number of parts under the CRP prior is

[1] Available at www.cs.brown.edu/ sghosh

roughly $\alpha \log N$; we set $\alpha = 2$ so that the expected number of mesh-crp parts is similar to the number of parts discovered by the mesh-ddcrp. To exploit bilateral symmetry, for all methods we only segment the right half of each mesh. The resulting segmentation is then reflected onto the left half.

4.2.6.3 Part Discovery and Motion Prediction

We first consider the synthetic Tosca dataset [120], and separately analyze the Centaur (six poses) and Horse (eight poses) meshes. These meshes contain about 31,000 and 38,000 triangular faces, respectively. Figure 4.3 displays the segmentations of the Tosca meshes inferred by mesh-ddcrp. The inferred parts largely correspond to groups of mesh faces which undergo similar transformations.

Figure 4.4 displays the results produced by the ddCRP, as well as our baseline methods, on the human mesh data. Qualitatively, the segmentations produced by mesh-ddcrp correspond to our intuitions about the body. Note that in addition to capturing the head and limbs, the segmentation successfully segregates distinctly moving small regions such as knees, elbows, shoulders, biceps, and triceps. In all, the mesh-ddcrp detects 20 distinctly moving parts for one half of the body.

We now introduce a quantitative measure of segmentation quality: segmentations are evaluated by their ability to explain the articulations of test meshes with novel shapes and poses. Given a collection of T test meshes Y_t with corresponding reference meshes X_{b_t}, and a candidate segmentation into K parts, we compute

$$\mathcal{E} = \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} ||Y_{tk} - A^{*}_{tk} X_{b_t k}||_2. \tag{4.17}$$

Here, A^{*}_{tk} is the least squares estimate of the single affine transformation responsible for mapping $X_{b_t k}$ to Y_{tk}. Note that Equation (4.17) is trivially zero for a degenerate solution wherein each mesh face is assigned to its own part. However, segmentations of similar resolution may safely be compared using Equation (4.17), with lower errors corresponding to better segmentations.

On our test set of human meshes, the mesh-ddcrp model produces an error of $\mathcal{E} = 1.39$ meters, which corresponds to sub-millimeter accuracy when normalized

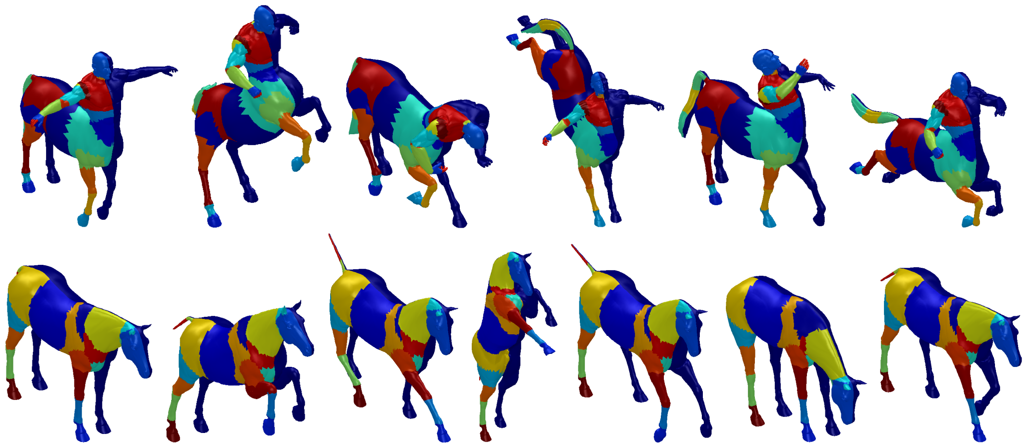


Figure 4.3: Segmentations produced by mesh-ddcrp on synthetic Tosca meshes [120]. The first mesh in each row displays the chosen reference mesh. For illustration, we have only segmented the right half of each mesh.

by the number of faces. Figure 4.4 displays a plot comparing the errors achieved by the different methods. Mesh-ddcrp is significantly better than all other methods, including for settings of K which allocate 50% more parts to competing approaches, according to a Wilcoxon's signed rank test (5% significance level).

Next, we demonstrate the benefits of sharing information among differently shaped bodies. We selected an illustrative articulated pose for each of the two training subjects in addition to their respective reference poses (Figure 4.4). The chosen poses either exhibit upper or lower body deformations, but not both. The meshes were then segmented both independently for the two subjects and jointly sharing information across subjects. Figure 4.5 demonstrates that the independent segmentations exhibit both undersegmented (legs in the first set) and oversegmented (head in the second) parts. However, sharing information among subjects results in parts which correspond well with physical human bodies. Note that with only two articulated poses, we are able to generate meaningful segmentations in about an hour of computation. This data-limited scenario also demonstrates the benefits of the ddCRP prior: as shown in Figure 4.5, the parts extracted by mesh-crp are “patchy”, spatially disconnected, and physically implausible.

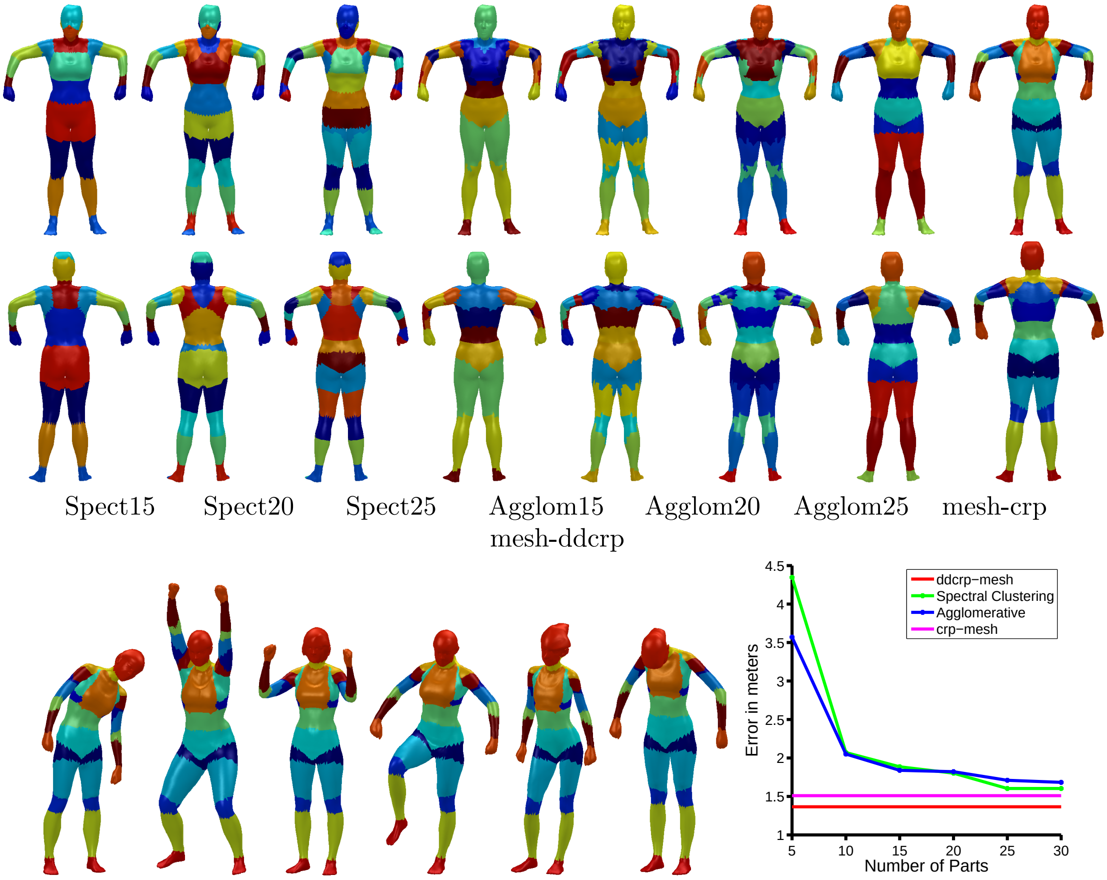


FIGURE 4.4: *Top two rows (left to right):* Segmentations produced by spectral and agglomerative clustering with 15, 20, and 25 clusters respectively, followed by the mesh-crp and mesh-ddcrp segmentations. *Bottom row:* Test set results. We display mesh-ddcrp segmentations for several test meshes, and quantitatively compare methods.

4.2.7 Bilateral Symmetry

In the prequel, we dealt with bilateral symmetry exhibited by human bodies, and more generally by symmetric objects by segmenting a half of the object and projecting the resulting labels across the axis of symmetry. As we have seen such a procedure produces reasonable results. However, it prevents us from modeling any variance in the deformations of the symmetric halves and forces both halves to the same part, giving rise to potentially suboptimal partitions. An obvious alternative would be to endow each symmetric half with its own link random variable allowing it to choose its part membership given the observed deformations. However, such a

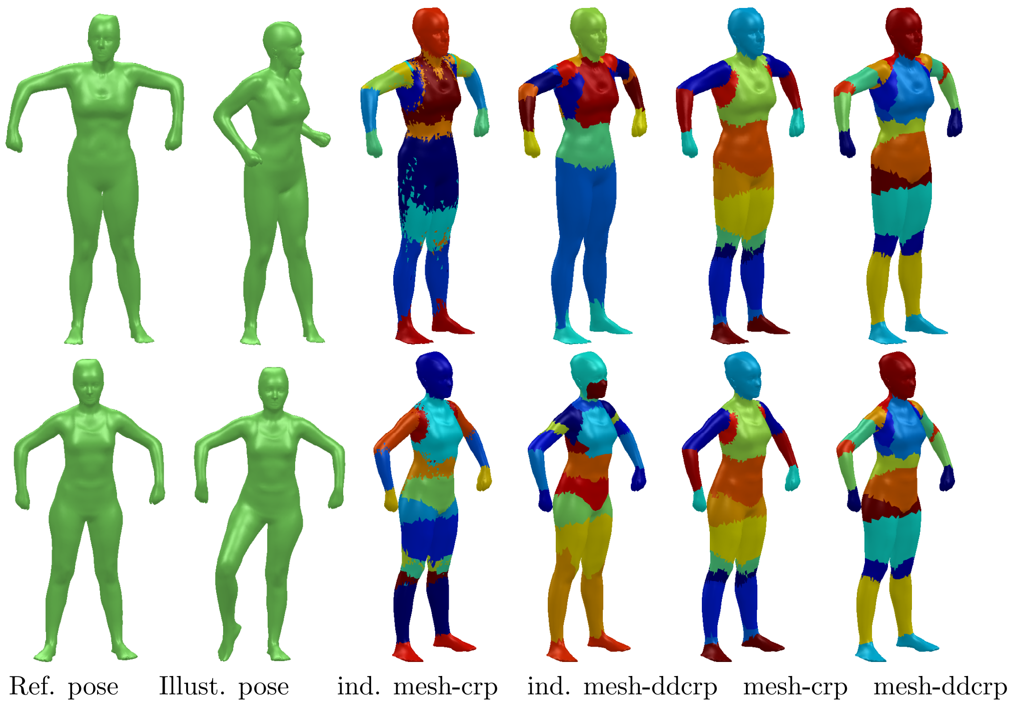


FIGURE 4.5: Impact of sharing information across bodies with varying shapes. The two rows correspond to the training subjects. Each row displays the reference pose, an illustrative articulated pose, mesh-crp and mesh-ddcrp segmentations produced by independently segmenting the pair of poses of each individual, and mesh-crp and mesh-ddcrp segmentations produced by jointly segmenting the chosen poses from both subjects.

model doubles the latent variables involved and thus requires much longer sampler runs and poses a significant computational challenge.

To alleviate such concerns, here we propose an alternate procedure. First, we extend the model presented in Section 4.2.3 to explicitly account for deformations observed in both symmetric halves, without increasing the number of associated latent varaibles. The allocation of triangles to parts and sampling of pose specific transformations and noise variables proceeds as before. However, given parts and transformations, we now sample the observed location of each pose triangle relative to its corresponding reference triangle in both symmetric halves, $y^l_{ji} \sim \mathcal{N}(A_{jz_i}x^l_{b_{ji}}, \Sigma_{jz_i})$ and $y^r_{ji} \sim \mathcal{N}(A_{jz_i}x^r_{b_{ji}}, \Sigma_{jz_i})$. Here the superscripts l and r refer to the triangles associated with the left and right symmetric halves. The overall extended model is then described as follows,

$$p(\mathbf{Y}, c, A, \Sigma \mid \mathbf{x}, b, D, \alpha, f, \eta) = p(c \mid D, f, \alpha)$$
$$\prod_{j=1}^{J} \left[\prod_{k=1}^{K(c)} p(A_{jk}, \Sigma_{jk} \mid \eta) \right] \left[\prod_{i=1}^{N} \prod_{h \in \{l,r\}} \mathcal{N}(y_{ji}^{h} \mid A_{jz_i} x_{b_{ji}}^{h}, \Sigma_{jz_i}) \right]. \quad (4.18)$$

Inference in the extended model can be performed using the Gibbs sampling algorithm presented in Section 4.2.5 after suitably replacing the likelihood terms of the original model with those from the extended model. After inferring part assignments, we perform a post-hoc test to ascertain whether two symmetric halves should share a common part. For every part k, we test whether the likelihood of assigning the symmetric halves to the same part exceeds the likelihood of assigning them to independent parts, $\frac{p(Y_{jk}^{l} \cup Y_{jk}^{r} \mid X_{b_{jk}}^{l} \cup X_{b_{jk}}^{r}, \eta)}{p(Y_{jk}^{l} \mid X_{b_{jk}}^{l}, \eta) p(Y_{jk}^{r} \mid X_{b_{jk}}^{r}, \eta)} > 1$. If the test succeeds then the inferred parts are retained, else they are split in two. We find that this procedure leads to intuitive results (Figure (4.6)). Parts such as the head and neck are never split, while arms, legs and breasts split into distinctly deforming parts.

4.2.8 Large Scale Human Studies

We further validate the generalizability of our conclusions with a significantly larger experiment. We acquired 1732 meshes from 78 human (both male and female) subjects of varying body types and in diverse poses. A subset of these meshes are depicted in Figure (4.6).

The availability of a large number of male and female meshes allows us to explore whether there were systematic differences in the parts discovered for the different sexes. We perform separate analyses of the male and female meshes by running 5 independent MCMC chains for 250 iterations each and selected the MAP sample. We then perform a hill climbing operation, wherein each mesh face selects the most likely link according to its conditional distribution, to the closest mode. Symmetry is accounted for using the methods described in Section 4.2.7. The resulting mesh decompositions are shown in Figure (4.6). We find that the discovered segmentations are largely consistent with those from the smaller dataset.

We also find that both male and female meshes exhibit similar deformations and provide no evidence of systematic biases between sexes.

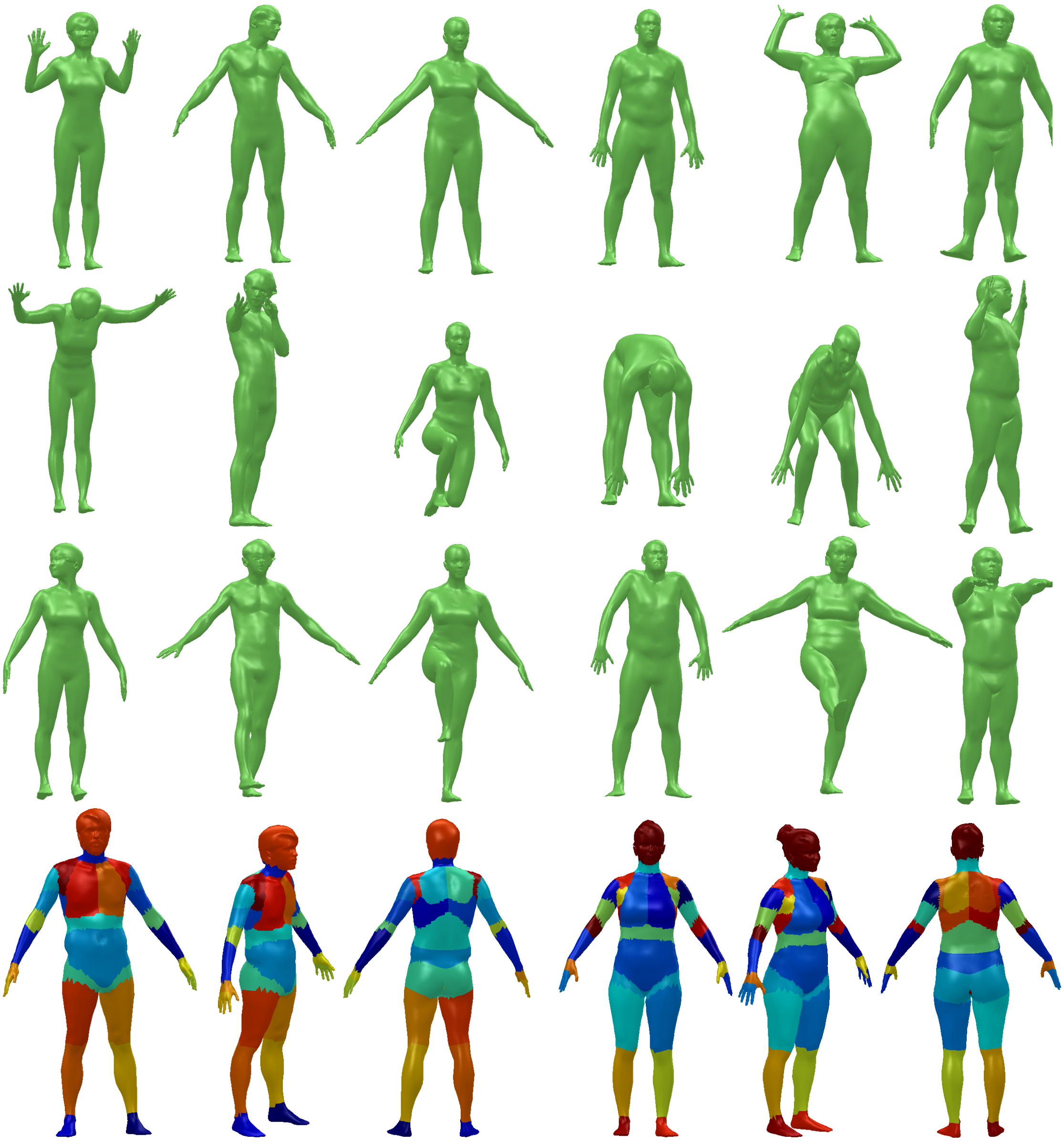


FIGURE 4.6: A large dataset of meshes acquired from 78 human subjects. The subjects exhibit diverse body shapes and poses. The last row depicts the parts discovered from observed deformations of male and female bodies. The male and female meshes were analyzed separately.

4.3 Discussion

Adapting the ddCRP to collections of 3D meshes, we have developed an effective approach for the discovery an unknown number of parts underlying articulated object motion. Unlike previous methods, our model guarantees that parts are spatially connected, and uses transformations to model instances with potentially varying body shapes. Via a novel application of matrix normal-inverse-Wishart priors, our sampler analytically marginalizes transformations for improved efficiency. While we have modeled part motion via affine transformations, future work should explore more accurate Lie algebra characterizations of deformation manifolds [121].

Experiments with a moderately large collection of real human body poses provide strong quantitative evidence that our approach produces state-of-the-art segmentations with many potential applications.

Chapter 5

Hierarchical Partitions of Non Exchangeable Data

In this chapter we develop hierarchical versions of the distance dependent Chinese restaurant process presented in the previous chapter. Such generalizations are useful for shared analysis of multiple groups of related but distinct data, such as collections of images, documents, time series. In such cases, it is often useful to share information across groups. For instance, consider a video sequence with objects spanning several frames. Segments representing these objects must then be shared among the frames. Additionally, the segments should be of similar shape and size and exhibit coherent motion across frames. The *hierarchical ddCRP* (hddCRP), discussed in this chapter, captures these desiderata.

The hddCRP captures local relationships among data instance like the ddCRP, but also uses affinities among latent clusters to extract further global dependencies. After an initial ddCRP partitioning, local clusters are grouped via additional links that depend on a user-specified measure of cluster similarity. This framework allows the hddCRP to model relationships that depend on *aggregate* properties of clusters such as size and shape, which may be difficult to capture with likelihoods alone. Given arbitrary cluster and data affinity functions, which need not arise from true distance metrics, the hddCRP defines a valid joint probability distribution on partitions.

5.1 Hierarchical Distance Dependent Clusters

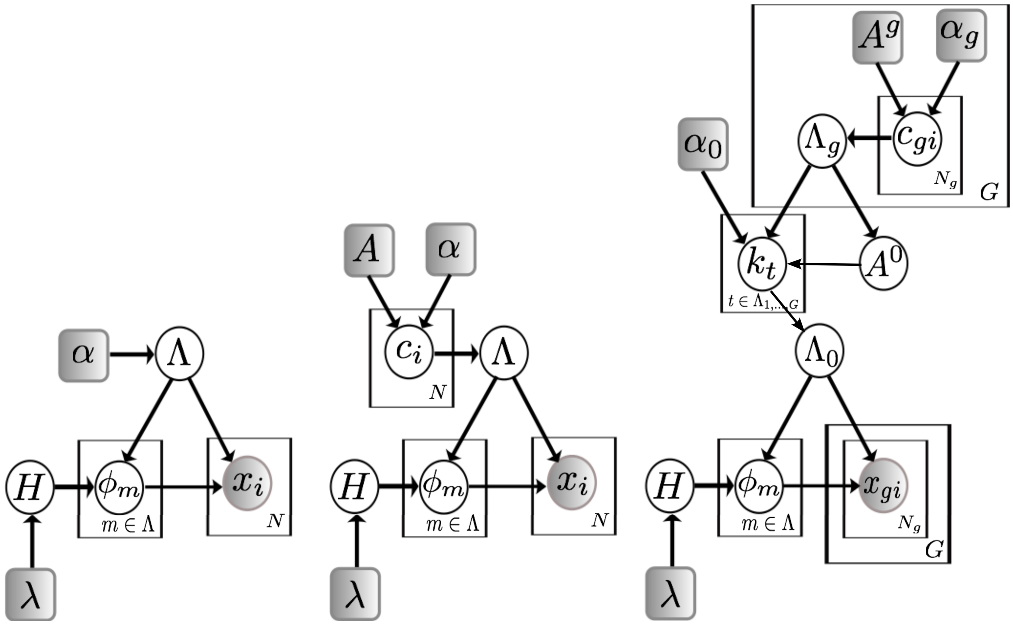


FIGURE 5.1: Graphical model representations for the CRP and ddCRP mixture models and the hddCRP hierarchical mixture model. In the CRP mixture, a partition is sampled from a CRP $\Lambda \sim \text{CRP}(\alpha)$; each component of the partition m is endowed with a parameter ϕ_m from a base distribution H and $x_i \sim \phi_m$ for $i \in m$. The ddCRP mixture replaces the prior over partitions with a ddCRP. Data links are sampled according to $c_i \sim p(c_i \mid \alpha, A)$, a connected components operation then generates the partition Λ. The hddCRP model first samples partitions Λ_g from group specific ddCRPs. Cluster links k_t, $t \in \Lambda_{1:G}$ are sampled from a cluster level ddCRP, $k_t \sim p(k_t \mid \alpha_0, A^0(\mathbf{c}))$. Connected components of the cluster links define a partition of the dataset Λ_0.

In Chapter 4, we noted that the distance-dependent CRP [108] defines a distribution over partitions indirectly via distributions over links between data instances. A data point i has an associated link variable c_i which links to another data instance j, or itself, according to the following distribution:

$$p\,(c_i = j \mid A, \alpha) \propto \begin{cases} A_{ij} & i \neq j, \\ \alpha & i = j. \end{cases} \tag{5.1}$$

The *affinity* $A_{ij} = f(d(i,j))$ depends on a user-specified *distance* $d(i,j)$ between pairs of data points, and a monotonically decreasing *decay function* $f(d)$ which makes links to nearby data more likely. The resulting link structure induces a partition, where two data instances are assigned to the same cluster if and only

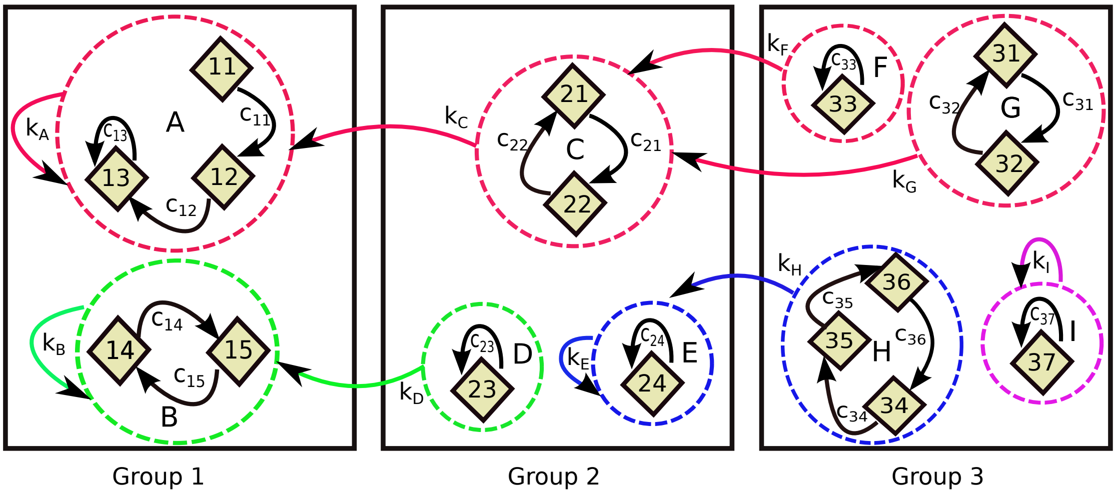


FIGURE 5.2: An example link variable configuration for a hierarchical ddCRP model of three groups (rectangles). Observed data points (customers, depicted as diamonds) link to other data points in the same group (black arrows), producing local clusters (dashed circles, labeled A through I). Cluster links (colored arrows) then join clusters to produce (in this case, four) global mixture components.

if one is reachable from the other by traversing the link edges. Larger self-affinity parameters α favor partitions with more clusters.

5.1.1 The Hierarchical ddCRP

We propose a novel generative model that applies the ddCRP formalism twice, first for clustering data within each group into local clusters, and then for coupling the local clusters across groups. Like the ddCRP, our hddCRP defines a valid distribution over partitions of a dataset. It places higher probability mass on partitions that group nearby data points into latent clusters, *and* couple similar local clusters into global components. Examples of these data and cluster links are illustrated in Figure 5.2.

Consider a collection of G groups, where group g contains N_g observations. We denote the i^{th} data point of group g by x_{gi}, and the full dataset by $\mathbf{x}$. The data link variable c_{gi} for x_{gi} is sampled from a group-specific ddCRP:

$$p(c_{gi} = gj \mid \alpha_g, A^g) \propto \begin{cases} A^g_{ij} & i \neq j, \\ \alpha_g & i = j. \end{cases} \tag{5.2}$$

At this first level of link variables, we set the probability of linking observations in different groups to zero. The connected components of the links $c_g = \{c_{gi} \mid i = 1, \ldots, N_g\}$ then determine the local clustering for group g.

Data links $\mathbf{c} = \{c_1, \ldots, c_G\}$ across all groups divide the dataset into group-specific local clusters $T(\mathbf{c})$. The hddCRP then associates each cluster $t \in T(\mathbf{c})$ with a cluster link k_t drawn from a global ddCRP distribution:

$$p(k_t = s \mid \alpha_0, A^0(\mathbf{c})) \propto \begin{cases} A^0_{ts}(\mathbf{c}) & t \neq s, \\ \alpha_0 & t = s. \end{cases} \tag{5.3}$$

Here α_0 is a global self-affinity parameter, and $A^0(\mathbf{c})$ is the set of pairwise affinities between the elements of $T(\mathbf{c})$. We let $A^0_{ts}(\mathbf{c}) = f_0(d_0(t, s, \mathbf{c}))$, where $d_0(t, s, \mathbf{c})$ is a "distance" based on arbitrary properties of clusters t and s, and $f_0(d_0)$ a decreasing decay function. The connected components of $\mathbf{k} = \{k_t \mid t \in T(\mathbf{c})\}$ then couple local clusters into global components shared across groups. Let z_{gi} denote the global component associated with observation i in group g, and $\mathbf{z} = \{z_{gi} \mid g = 1, \ldots, G; i = 1, \ldots, N_g\}$. Data instances x_{gi} and x_{hj} are clustered ($z_{gi} = z_{hj}$) if and only if they are reachable via some combination of data and cluster links.

Given this partition structure, we endow component m with likelihood parameters $\phi_m \sim G_0(\lambda)$, and generate observations $x_{gi} \sim p(x_{gi} \mid \phi_{z_{gi}})$. Let $M(\mathbf{c}, \mathbf{k})$ equal the number of global components induced by the cluster links $\mathbf{k}$ and data links $\mathbf{c}$. Because data links $\mathbf{c}$ are conditionally independent given $A^{1:G}$, and cluster links $\mathbf{k}$ are conditionally independent given $\mathbf{c}$ and the cluster affinities $A^0(\mathbf{c})$, the hddCRP joint distribution on partitions and observations factorizes as follows:

$$\begin{aligned} p(\mathbf{x}, \mathbf{k}, \mathbf{c} \mid \alpha_{1:G}, \alpha_0, A^{1:G}, A^0, \lambda) = {} & \prod_{m=1}^{M(\mathbf{c},\mathbf{k})} p(x_{\mathbf{z}=m} \mid \lambda) \\ & \prod_{g=1}^{G} \prod_{i=1}^{N_g} p(c_{gi} \mid \alpha_g, A^g) \prod_{k_t \in \mathbf{k}} p(k_t \mid \mathbf{c}, \alpha_0, A^0(\mathbf{c})) \end{aligned} \tag{5.4}$$

The set of data in component m is denoted by $x_{\mathbf{z}=m}$, and

$$p(x_{\mathbf{z}=m} \mid \lambda) = \int \prod_{gi \mid z_{gi}=m} p(x_{gi} \mid \phi_m) \, dH(\phi_m \mid \lambda), \tag{5.5}$$

where λ are hyperparameters specifying the prior distribution H. Our inference algorithms assume this integral is tractable, as it always is when an exponential family likelihood is coupled with an appropriate conjugate prior. We emphasize that for arbitrary data and cluster affinities, the sequential hddCRP generative process defines a valid joint distribution $p(\mathbf{x}, \mathbf{k}, \mathbf{c}) = p(\mathbf{c})p(\mathbf{k} \mid \mathbf{c})p(\mathbf{x} \mid \mathbf{k}, \mathbf{c})$.

5.1.2 Related Models

The hddCRP subsumes several recently proposed hierarchical extensions to the ddCRP, as well as the HDP itself, by defining appropriately restricted data affinities and local cluster affinities. Blei and Frazier [108] show that the CRP is recovered from the ddCRP by arranging data in an arbitrary sequential order, and defining affinities as

$$A_{ij} = \begin{cases} 1 & \text{if } i < j, \\ 0 & \text{if } i > j. \end{cases} \tag{5.6}$$

Data points link to all previous observations with equal probability, and thus the probability of joining any existing cluster is proportional to the number of other data points already in that cluster. The probability of creating a new cluster is proportional to the self-connection weight α. The resulting distribution on partitions can be shown to be invariant to the chosen sequential ordering of the data, and thus the standard CRP is *exchangeable* [122].

5.1.2.1 Hierarchical Chinese Restaurant Process (hCRP)

The hCRP representation of the HDP, which Teh et al. [123] call the “Chinese restaurant franchise”, is recovered from the hddCRP by first defining group-specific affinities as in Eq. (5.6). We then arrange local clusters t (tables, in the CRF metaphor) sequentially with affinities

$$A^0_{ts}(\mathbf{c}) = \begin{cases} 1 & \text{if } t < s, \\ 0 & \text{if } t > s. \end{cases} \tag{5.7}$$

Just as the two-level hCRP arises from a sequence of CRPs, the hddCRP is defined from a sequence of two ddCRP models.

5.1.2.2 Distance Dependent Chinese Restaurant Franchise

An alternate approach to capturing group-specific metadata uses a standard CRP to locally cluster data, but then uses the group labels to define affinities between clusters. Kim and Oh [124] use this model to learn topic models of time-stamped documents. By constraining cluster affinities to depend on group labels, but not properties of the data assigned to within-group clusters, inference is simplified.

5.1.2.3 Naive Hierarchical ddCRP (naive-hddCRP)

The image segmentation model of Ghosh et al. [52] clusters data within each group via a ddCRP based on an informative distance (in their experiments, spatial distance between image pixels). A standard CRP, as in the upper level of the HDP, is then used to combine these clusters into larger segments. The upper level CRP could either be expressed through cluster links with sequential affinities (Equation (5.7)) or directly through a CRP. In the direct representation clusters sample global component memberships $k_t \sim \text{CRP}(\alpha_0)$[1] directly instead of sampling links to other clusters. The absence of cluster links substantially simplifies inference for this special case.

5.2 Markov Chain Monte Carlo Inference

The posterior distribution over the data and cluster links $p(\mathbf{c}, \mathbf{k} \mid \mathbf{x}, \alpha_{1:G}, \alpha_0, A^{1:G}, A^0, \lambda)$ is intractable, and we thus explore it via a Metropolis-Hastings MCMC method. Our approach generalizes the non-hierarchical ddCRP Gibbs sampler of Blei and

[1] We are slightly abusing notation here. Here k_t are global component memberships instead of cluster links.

Frazier [108], which iteratively samples single data links conditioned on the observations and other data links. Evolving links lead to splits, merges, and other large changes to the partition structure. In the hddCRP, local clusters belong to global components, and these component memberships must be sampled as well.

5.2.1 Markov Chain State Space

The number of possible non-empty subsets (clusters) of N data points is $2^N - 1$. The state space of our Markov chain consists of the data links $\mathbf{c}$, and the set of *all* possible cluster links $\mathcal{K}$, one for each candidate non-empty cluster. For instance, given three observations $\{h, i, j\}$ the set of non-empty subsets is $\mathcal{T} = \{[h], [i], [j], [hi], [ij], [jh], [hij]\}$, and the corresponding set of possible cluster links is $\mathcal{K} = \{k_h, k_i, k_j, k_{hi}, k_{ij}, k_{jh}, k_{hij}\}$, where $|\mathcal{K}| = 2^3 - 1$.

For any configuration of $\mathbf{c}$, a strict subset of $\mathcal{T}$ will have data associated with it. We call this the *active set*. For instance, if $c_h = h, c_i = i, c_j = j$, then only the clusters $\{[h], [i], [j]\}$ and the corresponding links $\{k_h, k_i, k_j\}$ are active. Given $\mathbf{c}$, we split $\mathcal{K}$ into the active set $\mathbf{k}$, and the remaining inactive cluster links $\tilde{\mathbf{k}} = \mathcal{K} \setminus \mathbf{k}$. We account for the inactive clusters by augmenting $A^0(\mathbf{c})$ as follows:

$$\tilde{A}^0(\mathbf{c}) = \begin{bmatrix} A^0(\mathbf{c}) & \mathbf{0} \\ \mathbf{0} & \alpha_0 \mathbf{I} \end{bmatrix}. \tag{5.8}$$

Here, we have sorted the links so that affinities among the active clusters are listed in the upper-left quadrant of $\tilde{A}^0(\mathbf{c})$. As indicated by the identity matrix $\mathbf{I}$, inactive clusters have zero affinity with all other clusters, and link to themselves with probability one. Under this augmented model, the joint probability factorizes as follows:

$$\begin{aligned} p(\mathbf{x}, \mathbf{k}, \tilde{\mathbf{k}}, \mathbf{c}) &= p(\mathbf{c}) p(\mathbf{k} \mid \mathbf{c}) p(\tilde{\mathbf{k}} \mid \mathbf{c}) p(\mathbf{x} \mid \mathbf{c}, \mathbf{k}, \tilde{\mathbf{k}}) \\ &= p(\mathbf{c}) p(\mathbf{k} \mid \mathbf{c}) p(\tilde{\mathbf{k}} \mid \mathbf{c}) p(\mathbf{x} \mid \mathbf{c}, \mathbf{k}) \\ &= p(\mathbf{x}, \mathbf{k}, \mathbf{c}) p(\tilde{\mathbf{k}} \mid \mathbf{c}). \end{aligned} \tag{5.9}$$

Here, we have recovered the joint distribution of Eq. (5.4) because given $\mathbf{c}$, the observations $\mathbf{x}$ are conditionally independent of the inactive links $\tilde{\mathbf{k}}$. Crucially,

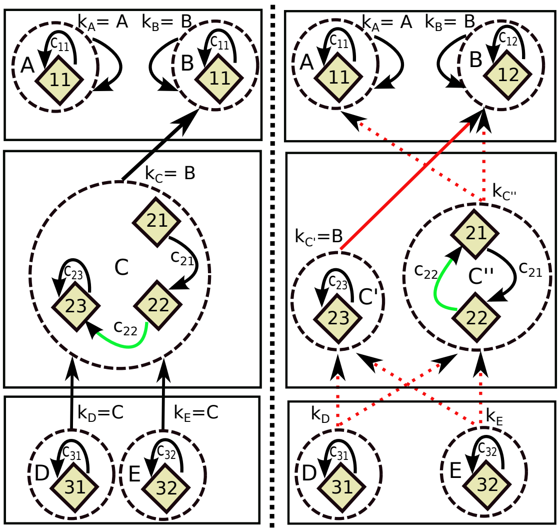


FIGURE 5.3: Illustration of changes induced by a data link proposal. Changing c_{22} (in the left configuration) splits cluster C into two clusters C' and C''. The cluster links associated with C (shown in red) must also be resampled. The MH step of the sampler proposes a joint configuration of the links $\{c_{22}, k_{C'}, k_{C''}, k_D, k_E\}$. The dashed red arrows illustrate the possible values the resampled cluster links could take. A single data link can create large changes to the partition structure, with local clusters splitting or merging, and groups of clusters shifting between components.

because inactive cluster links have no uncertainty, we must only explicitly represent the active clusters at each MCMC iteration.

As the Markov chain evolves, clusters are swapped in and out of the active set. Although the number of active clusters varies with the state of the chain, the dimensionality of the augmented state space $(\mathbf{c}, \mathbf{k}, \tilde{\mathbf{k}})$ remains constant, allowing us to ignore complications that arise when dealing with chains whose state spaces have varying dimensionality. In particular, we employ standard *Metropolis-Hastings* (MH) proposals to change data and cluster links, and need not resort to reversible jump MCMC [125].

5.2.2 Sampler Description

In samplers previously developed for the hCRP [123] and the naive-hddCRP [52], local clusters directly sample their global component memberships. However for the hddCRP, cluster links indirectly determine global component memberships.

This complicates inference, as any change to the cluster structure necessitates coordinated changes to cluster links. As illustrated in Figure 5.3, consider the case where a data link proposal causes a cluster to break into two components. The new cluster must sample a cluster (outgoing) link, and cluster links pointing to the old cluster (incoming links) must be divided among the newly split clusters. Thus, we use a MH proposal to jointly resample data and affected cluster links. After cycling through all data links $\mathbf{c}$, we use a Gibbs update to resample the cluster links $\mathbf{k}$. Algorithm 1 summarizes our procedure.

Algorithm 1: Hierarchical ddCRP sampler

```
For data instance $i \in \{1 \dots N_G\}$ jointly propose data and affected cluster links
$\{\mathbf{c}^*, \mathbf{k}^*\} \longleftarrow \text{ProposeLinks}(\mathbf{x}, \mathbf{k}, \mathbf{c}, \alpha_{1:G}, A^{1:G}, \alpha_0, A^0(\mathbf{c}))$.
Evaluate the proposal according to the Metropolis Hastings acceptance
probability $a(\{\mathbf{c}^*, \mathbf{k}^*\}, \{\mathbf{c}, \mathbf{k}\})$. If the proposal is accepted, $\{\mathbf{c}^*, \mathbf{k}^*\}$ becomes the
next state. If the proposal is rejected, the original configuration is retained.
For clusters $t \in T(\mathbf{c})$ resample cluster links via a Gibbs update:
$k_t \sim p(k_t \mid \mathbf{k}_{-t}, \mathbf{c}, \mathbf{x}, \alpha_0, A^0(\mathbf{c}))$.
```

5.2.2.1 Link Proposal Distributions

We now describe the algorithm for jointly proposing data and affected cluster links in more detail. To simplify the exposition, we focus on a particular group g and denote c_{gi} as c_i. Let the current state of the sampler be $\mathbf{k}(\mathbf{c})$ and $\mathbf{c} = \{\mathbf{c}_{-i}, c_i = j\}$, so that i and j are members of the same cluster t_{ij}. Let $\mathcal{K}_{t_{ij}} = \{k_s \mid k_s = t_{ij}, s \neq t_{ij}\}$ denote the set of other clusters linking to t_{ij}.

Split? To construct our link proposal, we first set $c_i = i$. This may split current cluster t_{ij} into two new clusters, in which case we let t_i denote the cluster containing data i, and t_j the cluster containing formerly linked data j. Or, the partition structure may be unchanged so that $t_i = t_{ij}$.

Incoming links $k_s \in \mathcal{K}_{t_{ij}}$ to a split cluster are independently assigned to the new clusters with equal probability:

$$q_{\text{in}}(\mathcal{K}_{t_{ij}}) = \prod_{k_s \in \mathcal{K}_{t_{ij}}} \left(\frac{1}{2}\right)^{\delta(k_s, t_i)} \left(\frac{1}{2}\right)^{\delta(k_s, t_j)} = \left(\frac{1}{2}\right)^{|\mathcal{K}_{t_{ij}}|}. \tag{5.10}$$

The current outgoing link is retained by one of the split clusters, $k_{t_j} = k_{t_{ij}}$. To allow likelihood-based link proposals, we *temporarily* fix the other cluster link as $k_{t_i} = t_i$.

Propose Link We compare two proposals for c_i, the ddCRP prior distribution $q(c_i) = p(c_i \mid \alpha, A)$, and a data-dependent "pseudo-Gibbs" proposal distribution:

$$q(c_i) \propto p(c_i \mid \alpha, A)\Gamma(\mathbf{x}, \mathbf{z}(\Delta), \lambda), \quad \Delta = (c_i, \mathbf{c}_{-i}, k_{t_i} = t_i, \mathbf{k}_{-t_i})$$
$$\Gamma(\mathbf{x}, \mathbf{z}(\Delta), \lambda) = \begin{cases} \dfrac{p(\mathbf{x}_{\mathbf{z}(\Delta)=m_a} \cup \mathbf{x}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)}{p(\mathbf{x}_{\mathbf{z}(\Delta)=m_a} \mid \lambda)p(\mathbf{x}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)} & \text{if } c_i \text{ merges } m_a,\, m_b, \\ 1 & \text{otherwise.} \end{cases} \tag{5.11}$$

The prior proposal, although naïve, can perform reasonably when A is sparse. The pseudo-Gibbs proposal is more sophisticated, as data links are proposed conditioned on both the observations $\mathbf{x}$ and the current state of the sampler. Our experiments in Sec. 5.3 show it is much more effective.

Merge? Let $c_i = j^*$ denote the new data link sampled according to either the ddCRP prior or Eq. (5.11). Relative to the reference configuration in which $c_i = i$, this link may either leave the partition structure unchanged, or cause clusters t_i and t_{j^*} to merge into t_{ij^*}. In case of a merge, the new cluster retains the current outgoing link $k_{t_{ij^*}} = k_{t_{j^*}}$, and inherits the incoming links $\mathcal{K}_{t_{ij^*}} = \mathcal{K}_{t_i} \cup \mathcal{K}_{t_{j^*}}$.

If a merge does not occur, but t_{ij} was previously split into t_i and t_j, the outgoing link $k_{t_j} = k_{t_{ij}}$ is kept fixed. For newly created cluster t_i, we then propose a corresponding cluster link k_{t_i} from its full conditional distribution:

$$q_{\text{out}}(k_{t_i}) = p(k_{t_i} \mid \alpha_0, A^0(\mathbf{c}), \mathbf{x}, \mathbf{k}_{-t_i}, \mathbf{c}). \tag{5.12}$$

Note that the proposal $c_i = j^*$ may leave the original partition unchanged if $c_i = i$ does not cause t_{ij} to split, and $c_i = j^*$ does not result in a merge. In this case, the corresponding cluster links are also left unchanged.

Accept or Reject Combining the two pairs of cases above, our overall proposal distribution equals

$$q(\mathbf{c}^*_; \mathbf{k}^* | \mathbf{c}, \mathbf{k}, \mathbf{x}) = \begin{cases} q(c_i^*) q_{\text{in}}(\mathcal{K}^*_{t_{ij}}) & \text{split, merge,} \\ q(c_i^*) & \text{no split, merge,} \\ q(c_i^*) q_{\text{out}}(k^*_{t_i}) q_{\text{in}}(\mathcal{K}^*_{t_{ij}}) & \text{split, no merge,} \\ p(c_i^* \mid \alpha, A) & \text{otherwise.} \end{cases} \tag{5.13}$$

Here, $\mathbf{c}^*$ and $\mathbf{k}^*$ denote the proposed values, which are then accepted or rejected according to the MH rule.

5.2.2.2 Metropolis-Hastings Acceptance Probability

The MH acceptance probability takes the following well known form,

$$a(\beta, \beta^*) = \min \left(1, \frac{p(\beta^*)}{p(\beta)} \frac{p(\mathbf{x} \mid \beta^*)}{p(\mathbf{x} \mid \beta)} \frac{q_{rev}(\beta \mid \beta^*, \mathbf{x})}{q_{fwd}(\beta^* \mid \beta, \mathbf{x})} \right), \tag{5.14}$$

where $\beta = \{\mathbf{c}, \mathbf{k}\}$ and we have dropped the hyper-parameters from the notation. The four cases in Equation (5.13) need to be considered. Here, we derive the acceptance ratio ρ_s for the split, no merge case. The acceptance ratios for the other cases follow analogously.

$$\rho_s(\beta, \beta^*) = \frac{p(\mathbf{x}, \beta^*)}{p(\mathbf{x}, \beta)} \frac{q_{rev}(\beta \mid \beta^*, \mathbf{x})}{q_{fwd}(\beta^* \mid \beta, \mathbf{x})} \tag{5.15}$$

Observe that the split, no merge move and the merge, no split moves are reverses of each other. Thus, from Equation (5.13) we have:

$$\rho_s(\beta, \beta^*) = \frac{p(\mathbf{x}, \beta^*)}{p(\mathbf{x}, \beta)} \frac{q(c_i)}{q(c_i^*) q_{\text{out}}(k^*_{t_i}) q_{\text{in}}(\mathcal{K}^*_{t_{ij}})} \tag{5.16}$$

First, considering the case where customer links are proposed from the prior $q(c_i) = p(c_i \mid \alpha, A)$ and expanding β, we have,

$$\rho_s(\beta, \beta^*) = \frac{(0.5)^{|L_{t_{i,i'}}|} p(\mathbf{c}^*) p(\mathbf{k}^* \mid \mathbf{c}^*) p(\mathbf{x} \mid \mathbf{k}^*, \mathbf{c}^*)}{p(\mathbf{c}) p(\mathbf{k} \mid \mathbf{c}) p(\mathbf{x} \mid \mathbf{k}, \mathbf{c})} \frac{p(c_i = j)}{p(c_i = j^*) p(k^*_{t_i} \mid \mathbf{x}, \mathbf{k}^*_{-t_i}, \mathbf{c}^*)} \tag{5.17}$$

The customer links cancel and the above equation simplifies as follows,

$$\begin{aligned} \rho_s(\beta, \beta^*) &= (0.5)^{|L_{t_{i,i'}}|} \frac{p(\mathbf{k}^* \mid \mathbf{c}^*) p(\mathbf{x} \mid \mathbf{k}^*, \mathbf{c}^*)}{p(\mathbf{k} \mid \mathbf{c}) p(\mathbf{x} \mid \mathbf{k}, \mathbf{c})} \frac{1}{p(k^*_{t_i} \mid \mathbf{x}, \mathbf{k}^*_{-t_i}, \mathbf{c}^*)}, \\ &= (0.5)^{|L_{t_{i,i'}}|} \frac{p(k^*_{t_i} \mid \mathbf{c}^*)}{p(k^*_{t_i} \mid \mathbf{x}, \mathbf{k}^*_{-t_i}, \mathbf{c}^*)} \frac{p(\mathbf{k}^*_{-t_i} \mid \mathbf{c}^*)}{p(\mathbf{k} \mid \mathbf{c})} \frac{p(\mathbf{x} \mid \mathbf{k}^*, \mathbf{c}^*)}{p(\mathbf{x} \mid \mathbf{k}, \mathbf{c})} \end{aligned} \tag{5.18}$$

The likelihood terms further simplify, yielding

$$\begin{aligned} \rho_s(\beta, \beta^*) &= \begin{cases} \tau \dfrac{p(\mathbf{x}_{z=m_a} \mid \lambda) p(\mathbf{x}_{z=m_b} \mid \lambda)}{p(\mathbf{x}_{z=m_a} \cup \mathbf{x}_{z=m_b} \mid \lambda)} & \text{if } c_i = j^* \text{ splits a component into } m_a \text{ and } m_b, \\ \tau & \text{otherwise,} \end{cases} \\ \tau &= \frac{(0.5)^{|L_{t_{i,i'}}|} p(k^*_{t_i} \mid \mathbf{c}^*)}{p(k^*_{t_i} \mid \mathbf{x}, \mathbf{k}^*_{-t_i}, \mathbf{c}^*)} \frac{p(\mathbf{k}^*_{-t_i} \mid \mathbf{c}^*)}{p(\mathbf{k} \mid \mathbf{c})} \ . \end{aligned} \tag{5.19}$$

Next, let us consider the pseudo-Gibbs proposals. When a proposed data link causes a component to split, the forward transition probability according to the pseudo-Gibbs proposal is,

$$\begin{aligned} q_{fwd}(\beta^* \mid \beta, \mathbf{x}) &= q(c_i^*) q_{\text{out}}(k^*_{t_i}) q_{\text{in}}(\mathcal{K}^*_{t_{ij}}) \\ &= \frac{1}{(0.5)^{|L_{t_{i,i'}}|} \mathcal{C}_i} p(c_i = j^* \mid \alpha, A) p(k^*_{t_i} \mid A^0(\mathbf{c}^*), \mathbf{x}, \mathbf{k}_{-t_i}), \end{aligned} \tag{5.20}$$

where $\mathcal{C}_i$ is the appropriate normalization constant for the discrete pseudo Gibbs proposal. The reverse move must cause two distinct components (m_a and m_b) to merge, yielding the following reverse transition probability,

$$\begin{aligned} q_{rev}(\beta \mid \beta^*, \mathbf{x}) &= q(c_i) \\ &= \frac{1}{\mathcal{C}_i} p(c_i = j \mid \alpha, A) \frac{p(\mathbf{x}_{z(\Delta)=m_a} \cup \mathbf{x}_{z(\Delta)=m_b} \mid \lambda)}{p(\mathbf{x}_{z(\Delta)=m_a} \mid \lambda) p(\mathbf{x}_{z(\Delta)=m_b} \mid \lambda)}. \end{aligned} \tag{5.21}$$

Together with Equation (5.16), this leads to the following acceptance ratio,

$$\begin{aligned}\eta_s(\beta, \beta^*) &= \frac{(0.5)^{|L_{t_{i,i'}}|} p(\mathbf{k}^*_{-t_i} \mid \mathbf{c}^*)}{p(\mathbf{k} \mid \mathbf{c})} \frac{p(k^*_{t_i} \mid \mathbf{c}^*)}{p(k^*_{t_i} \mid \mathbf{x}, \mathbf{k}^*_{-t_i}, \mathbf{c}^*)} \frac{p(\mathbf{x}_{z=m_a} \mid \lambda) p(\mathbf{x}_{z=m_b} \mid \lambda)}{p(\mathbf{x}_{z=m_a} \cup \mathbf{x}_{z=m_b} \mid \lambda)} \\ &\qquad \frac{p(\mathbf{x}_{z(\Delta)=m_a} \cup \mathbf{x}_{z(\Delta)=m_b} \mid \lambda)}{p(\mathbf{x}_{z(\Delta)=m_a} \mid \lambda) p(\mathbf{x}_{z(\Delta)=m_b} \mid \lambda)}, \\ &= \tau \frac{p(\mathbf{x}_{z=m_a} \mid \lambda)}{p(\mathbf{x}_{z=m_a} \cup \mathbf{x}_{z=m_b} \mid \lambda)} \frac{p(\mathbf{x}_{z(\Delta)=m_a} \cup \mathbf{x}_{z(\Delta)=m_b} \mid \lambda)}{p(\mathbf{x}_{z(\Delta)=m_a} \mid \lambda)}. \end{aligned} \tag{5.22}$$

When the proposed link does not cause a component to split into distinct components the prior and the pseudo Gibbs proposals are identical. Thus, the acceptance ratio for the split move under the pseudo Gibbs proposal is,

$$\rho_s^{pg}(\beta, \beta^*) = \begin{cases} \eta_s(\beta, \beta^*) & \text{if } c_i = j^* \text{ splits a component into } m_a \text{ and } m_b, \\ \tau & \text{otherwise.} \end{cases} \tag{5.23}$$

The acceptance ratio for the other moves are computed similarly and are available in the appendix.

5.2.2.3 Cluster Links Resampling Procedure

After having resampled the data and affected links, we resample all cluster links conditioned on data links using a simple Gibbs step. This is analogous to the original ddCRP sampler, cluster link k_t is sampled from,

$$p(k_t \mid \alpha_0, A^0(\mathbf{c}), \mathbf{x}, \mathbf{k}_{-t}) = \begin{cases} p(k_t \mid \alpha_0, A^0(\mathbf{c})) \dfrac{p(x_{z=m_a} \cup x_{z=m_b} \mid \lambda)}{p(x_{z=m_a} \mid \lambda)\, p(x_{z=m_b} \mid \lambda)} & \text{if } k_t \text{ merges } m_a \text{ and } m_b, \\ p(k_t \mid \alpha_0, A^0(\mathbf{c})) & \text{otherwise.} \end{cases} \tag{5.24}$$

5.2.2.4 Data link proposal comparisons

Intuitively, one would expect the prior proposals to be less effective than the data informed pseudo-Gibbs proposal. The intuition was confirmed in [126], where the authors demonstrated that the pseudo-Gibbs proposal more consistently reached higher probability states (Figure 5.4). In the remainder of this article, we focus solely on the pseudo-Gibbs proposal.

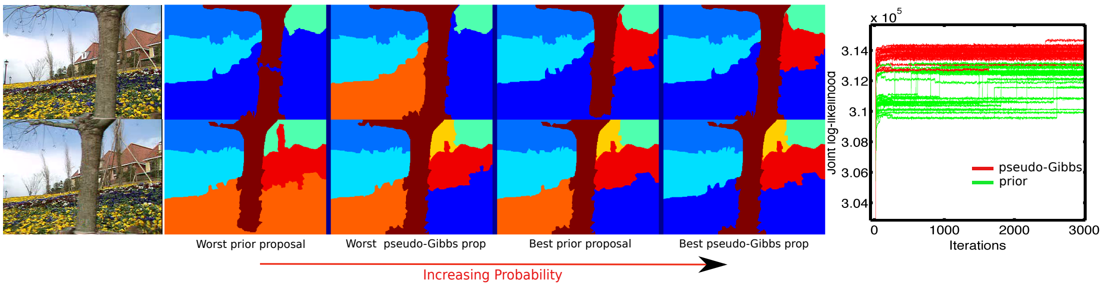


FIGURE 5.4: Data link proposal comparisons (reproduced from [126]). *Left:* Two frames from the "garden" video sequence, and partitions corresponding to the best and worst MAP samples using prior or pseudo-Gibbs proposals. *Right:* Joint log-likelihood trace plots for 25 trials of each proposal.

5.3 Experiments

The Hierarchical distance dependent Chinese restaurant process is a flexible statistical model that can be applied to several problems. In this section we explore its application to the tasks of activity and discourse segmentation.

5.3.1 Activity Discovery from Multiple Time Series

We consider the problem of analyzing collections of related time series with the goal of discovering shared commonalities among them. We restrict our attention to time series produced by motion capture sensors on the joints of people performing exercise routines. Each recording generates a multivariate time series that comprises of several locally coherent, simple dynamics that persist over a contiguous period of time and correspond to an exercise type (e.g., twists, jumping jacks and arm-circles). Here, we analyze motion capture recordings from multiple subjects, each performing a subset of a global set of exercises. By jointly analyzing these sequences we aim to discover the set of global exercises and their occurrences in each subject's motion capture stream.

5.3.1.1 Data

We analyze the motion capture recordings from the CMU MoCap database (http://mocap.cs.cmu.edu). Each motion capture sequence in this database consists of 64 measurements of human subjects performing various exercises. Following [127], we select 12 measurements deemed most informative for capturing gross motor behaviors: body torso position, neck angle, two waist angles, and a symmetric pair of right and left angles at each subjects shoulders, wrists, knees, and feet. Each recording thus provides a 12-dimensional time series. [127] provide a curated subset of the CMU MoCAP data set that contains six twelve dimensional sequences, three from two subjects each. In addition to having several exercise types in common this subset comes with human annotated ground truth labels allowing for easy quantitative comparisons across different models. We perform our experiments on this annotated subset.

5.3.1.2 Prior

We model the shared partition across MoCap sequences via the hddCRP. We use sequential ddCRPs to model individual time series. Each measurement within a MoCap sequence connects to others with probability

$$p(c_{gi} = gj \mid \alpha_g, A^g) \propto \begin{cases} \exp(-\frac{(i-j)}{N_g^\gamma}) & i > j, \\ 0 & i < j, \\ 1 & i = j. \end{cases} \tag{5.25}$$

Based on preliminary experiments we set $\gamma = \frac{1}{5}$. Series specific segments (or local clusters) then connect across time series with distances that correspond to a regular CRP prior (See Section 5.1).

5.3.1.3 Likelihood

Our data consists of six MoCap series. We denote by $x_{gi} \in \mathbb{R}^{D\times 1}; D = 12$ the measurement belonging to series g at time step i. We model the dynamics within a series using switching vector autoregressive (VAR) processes of order 1. Thus

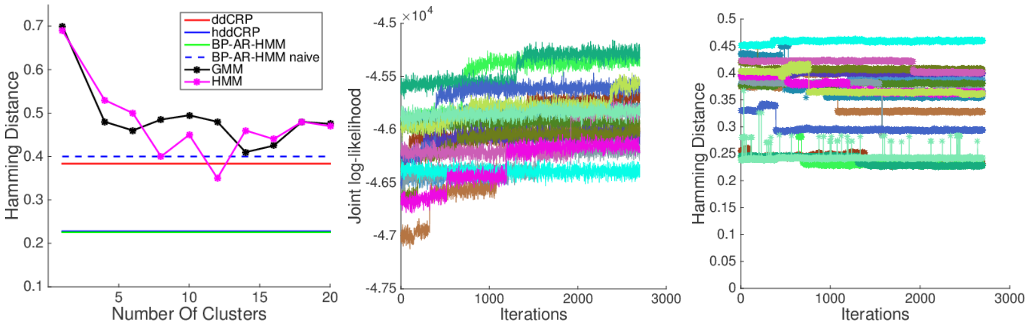


FIGURE 5.5: Motion capture segmentation. *Left:* Joint Log likelihood trace plots for 15 MCMC runs. *Right:* Corresponding normalized hamming distances achieved by the chains. In general, higher probability states correspond to lower hamming distances (and errors).

conditioned on the global component membership z_{gi} the measurement at time step i is modeled as follows:

$$x_{gi} = B_{z_{gi}} x_{gi-1} + \epsilon_{z_{gi}} \tag{5.26}$$

Further, we place a matrix normal inverse wishart (MNIW) prior on the auto regressive matrix B as follows:

$$\begin{aligned} \Sigma_{z_{gi}} \mid n_0, S_0 &\sim \text{IW}(n_0, S_0), \\ B_{z_{gi}} \mid M, \Sigma_{z_{gi}}, L &\sim \mathcal{MN}(M, \Sigma_{z_{gi}}, L), \\ \epsilon_{z_{gi}} &\sim \mathcal{N}(0, \Sigma_{z_{gi}}), \end{aligned} \tag{5.27}$$

where n_0 is the degrees of freedom, S_0 the scale matrix, M the mean dynamic matrix, and L along with $\Sigma_{z_{gi}}$ together control the covariance around M. In our experiments we set M to the identity matrix encoding our belief that in expectation the MoCap sequences locally exhibit simple random walk dynamics. Following [127], we set n_0 to $D + 2$ and S_0 to 0.5 times the empirical covariance of first differences of all observation sequences and L to $0.5\mathbf{I}_D$ where $\mathbf{I}_D$ is a D dimensional identity matrix.

5.3.1.4 Empirical Comparisons

We begin by exploring the benefits of the hierarchical ddCRP by comparing it with the ddCRP. The ddCRP model ignores sequence boundaries and segments a MoCap sequence created by stacking the six individual sequences together. We endow the ddCRP with the same likelihood model as the hddCRP (equation 5.27) and a prior identical to the local ddCRP models (equation 5.25) used by the hddCRP.

We also compare against the state of the art Beta process auto regressive hidden Markov model (BP-AR-HMM) presented in Fox et al. [127]. Additionally, we also compare against a Gaussian mixture model (GMM) and a Hidden Markov model (HMM) previously proposed [128] for activity clustering.

We benchmark performance using the normalized Hamming distance which is computed by measuring the fraction of time-steps where the inferred segmentation and the human annotated ground truth labels differ. Before computing the Hamming distance we find the optimal alignment of the estimated and true labels using the Hungarian algorithm.

We ran 15 randomly initialized hddCRP and ddCRP MCMC chains each for 3000 iterations. After discarding the first 10% of each chain to account for burn in, we selected the MAP sample from the remaining samples as our solution. The comparisons are presented in Figure 5.5. The hddCRP significantly outperforms the GMM and HMM baselines. It also performs much better than the ddCRP demonstrating the benefits of incorporating the hierarchy into the model. It achieves a normalized Hamming distance of 0.23 which is within noise of the BP-AR-HMM utilizing the most sophisticated data driven sampler developed in [127]. The BP-AR-HMM with a more naive sampler performs significantly worse. In contrast, the hddCRP allows for a collapsed sampler which works off the shelf requiring no additional tuning. A subset of the activities discovered by the hddCRP are visualized in Figure 5.6.

5.3.2 Discourse Segmentation

Next, we consider the problem of discourse segmentation. Given a collection of documents, the goal is to partition each document into a sequence of topically coherent non-overlapping discourse fragments. Previous work by Riedl and Biemann [129] found that sharing information across documents tends to produce

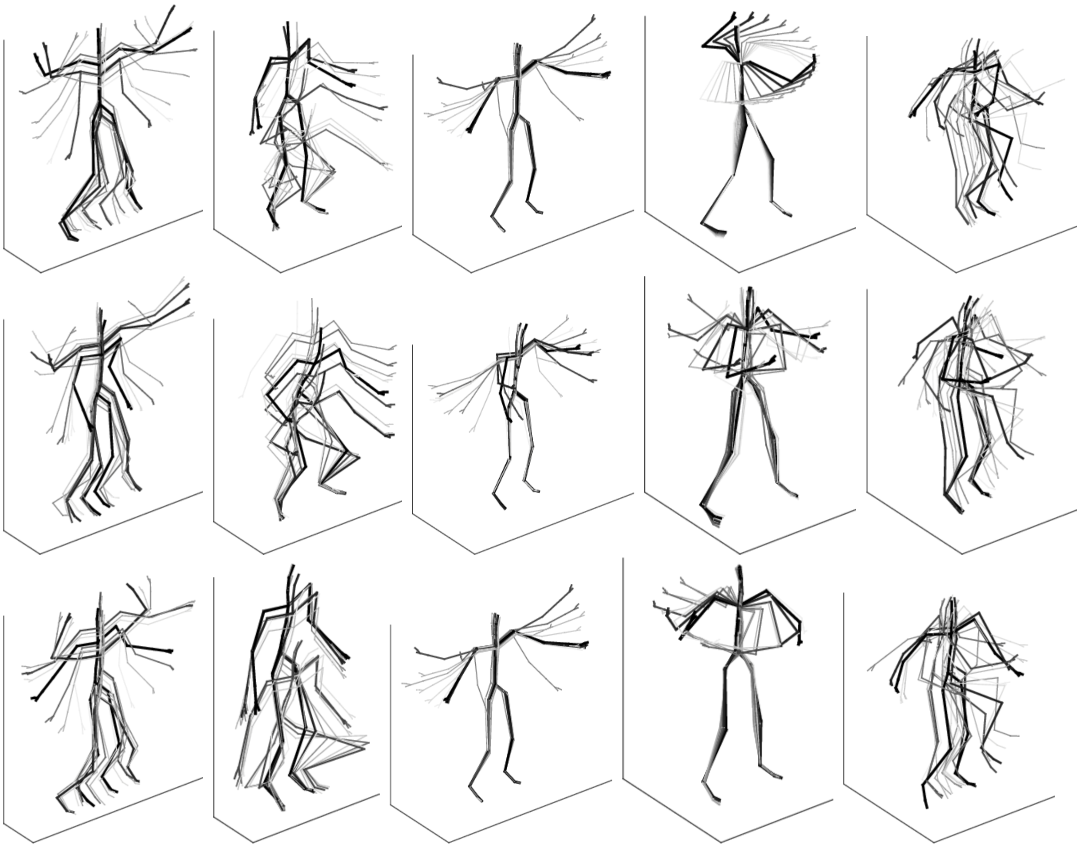


Figure 5.6: Illustrative examples of activities discovered by hddCRP – *left-to-right* jumping jacks, squats, arm circles, twists and knee raises. The model is robust to natural variability in activities arising from different subjects performing the activities.

better segmentations, motivating the development of several text segmentation algorithms that exploit document relationships.

5.3.2.1 Data

We conducted experiments on the *wikielements* dataset [130], which consists of 118 Wikipedia articles (at paragraph resolution) describing chemical elements. Although not explicitly made available in the dataset, each article corresponds to a chemical element characterized by its chemical properties and a unique location in the periodic table. Our distance-dependent models are capable of exploiting

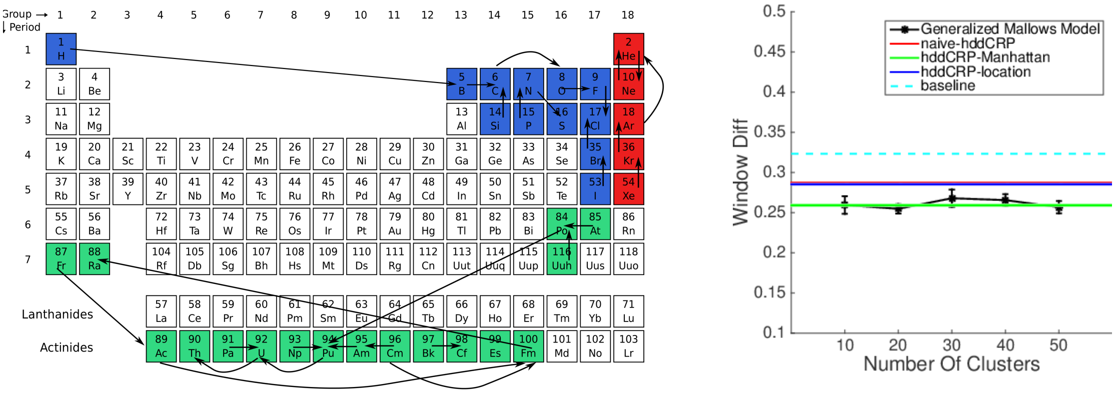


FIGURE 5.7: Discourse segmentation results on the *wikielements* dataset. *Left:* A partial visualization of the inferred customer links when clustering Wikipedia articles describing 118 chemical elements. The distance between articles equals the Manhattan distance between their locations in the periodic table. Only three of the nine discovered clusters have been visualized. *Right*: windowDiff scores and corresponding variances, achieved by competing methods. Lower scores indicate better performance. For the three hddCRP variants, the error bars are subsumed by the thickness of the plotted line.

this additional information to produce better discourse segmentations. As an illustration, consider the alternative problem of clustering articles. Figure 5.7 illustrates such a clustering where we leverage element properties by defining distances between documents as the Manhattan distance between corresponding element locations in the periodic table. The discovered clustering corresponds well with known element groupings. Discourse segmentation requires clustering the content describing documents, instead of the documents themselves. Nonetheless, we find that exploiting the periodic table location of each document's element leads to noticeable performance gains.

5.3.2.2 Prior

We experiment with three hddCRP priors capturing different intuitions about shared discourse structure across related documents. To encourage topic contiguity, all versions use data affinities that allow paragraphs to either link to themselves or to other paragraphs immediately preceding or succeeding them.

First, to capture the linguistic observation [131] that similar documents tend to present similar topics in similar orders, we consider a cluster level affinity function that bias clusters of paragraphs to connect to those that occur at similar locations within other documents. We refer to this model as location-hddCRP. We also define an affinity function (Manhattan-hddCRP) that captures the intuition that clusters are more likely to be shared among articles about similar elements. It models affinities between articles using the Manhattan distance between the corresponding element locations in the periodic table, modulated by a logistic decay function $f(d) = (1 + \exp(d))^{-1}$. Cluster affinities are defined as the affinity between the articles containing them. Further, the affinity between clusters in the same article is defined to be 0 for both models. Finally, as a baseline we also compare against the naive-hddCRP model.

5.3.2.3 Likelihood

Following previous work [130], we treat each article as a collection of paragraphs. Paragraph i in document g is represented as a histogram of words x_{gi}. Given the global component membership z_{gi}, x_{gi} is modeled as a Multinomial distribution with a symmetric Dirichlet prior.

$$x_{gi} \mid z_{gi} \sim \text{Mult}(\phi_{z_{gi}}) \ , \ \phi_{z_{gi}} \sim \text{Dir}(\lambda). \tag{5.28}$$

In our experiments, the hyper-parameter λ is set to 0.1 to encourage sparsity.

5.3.2.4 Empirical Comparisons

We benchmark our algorithms against the generalized Mallows model based text segmentation [130] approach and a naïve baseline that groups the entire dataset into one segment. We quantify performance using the windowDiff metric, which slides a window through the text incurring a penalty on discrepancies between the number of segmentation boundaries in the inferred segmentation and a gold standard segmentation. Lower windowDiff numbers indicate a better match with the ground truth.

Since the generalized mallow's model requires the number of clusters (K) to be pre-specified, we run the model with a number of different choices for K. Following the protocol presented in [130], we run five MCMC chains and collect the 10000^{th} sample from each chain. The mean windowDiff scores along with the associated variance, across the different cluster choices is summarized in Figure 5.7. For the hddCRP variants, we run 15 independent MCMC chains for 3000 iterations and select the 5 most probable samples across the chains. The results in Figure 5.7 report the mean windowDiff score (and variance) achieved by the different hddCRP variants. We observe that naive-hddCRP performs poorly, closely followed by location-hddCRP. This suggests that capturing structure between latent clusters is important, but ordering of discourse elements across documents is not pronounced in this collection of multi author wikipedia articles. Manhattan-hddCRP however performs quite well and is within noise of the *state-of-the-art* Mallows model for well chosen K and is superior for suboptimal choices of K. This significant improvement over naive-hddCRP suggests that modeling dependencies between latent clusters is important for discourse segmentation. For the collection of wikipedia chemical elements, similarity in content appears to be a stronger effect than similarity in location within the article.

5.4 Discussion

In this chapter we developed and investigated properties of the hierarchical distance dependent Chinese restaurant process, a versatile probabilistic model for shared clustering of groups of data exhibiting complex structure. We also designed effective MCMC algorithms for exploring the posterior over partitions induced by the hddCRP.

Applying the hddCRP to diverse domains is straightforward: one need only specify appropriate distance functions. The hierarchical ddCRP defines a valid joint probability distribution for any choice of affinities, which need not be metrics or have any special properties. Using temporal affinities, it produces *state-of-the-art* activity recognition results and leveraging distances based on paragraph order and element positions in the periodic table, it performs comparably to *state-of-the-art* textual discourse segmentation techniques.

Finally, we note that while our MCMC inference methods are highly effective for moderate-sized datasets, further innovations will be needed for computational scaling to very large datasets.

Chapter 6

Learning Distributions Over Partitions

The ddCRP and hddCRP models introduced in preceding chapters, specify distributions over partitions that are capable of modeling a wide variety of data dependencies through user specified affinity functions. Although, affinity functions are crucial for capturing domain specific information, designing functions that appropriately capture domain knowledge can be challenging. Moreover, the labor intensive design process often needs to be repeated for new application domains. As a result, previous applications [52, 108, 132, 133] have resorted to simple, intuitive affinity functions. However, it is unclear whether such functions are optimal or how functions for new domains may be designed without significant experimentation.

In this chapter, we alleviate such issues by developing covariate augmented models that express pairwise similarities between data points or clusters as functions of covariates. The modeler is no longer required to specify arbitrary affinity functions. Instead, she is only required to provide potentially *weak* cues encoding similarities between data instances and between latent clusters.

6.1 Covariate Augmented Models

We model pairwise affinities as weighted linear combinations of covariates modulated via monotonic *nonlinear* functions (f),

$$\begin{aligned} A_{ij} &= f(w_c^T \theta_{ij}^c), \qquad w_c \sim \mathcal{N}(0, \Psi_c), \\ A_{ts}^0 &= f(w_k^T \theta_{ts}^k), \qquad w_k \sim \mathcal{N}(0, \Psi_k). \end{aligned} \tag{6.1}$$

Here, θ_{ij}^c, θ_{ts}^k are user provided covariates encoding similarities between data instances i, j and latent clusters t, s. We endow the weights w_c and w_k with large variance, independent, zero mean Gaussian priors. Graphical model representations of the covariate augmented models are presented in Figure 6.1. Observe, that the affinity functions are now parametrized by weight vectors w_c and w_k. In the next section, we will develop algorithms for reliably learning these parameters from moderate sized collections of human annotated partitions. We also note that similar models [134] have recently been proposed in the literature. However, to the best of our knowledge, the challenging problem of learning from human annotations has not been previously addressed.

6.2 Loss Aware Learning

We consider the problem of learning weights $w = \{w_c, w_k\}$ given human annotated training partitions $Y = \{y_1 \dots y_D\}$. Here a partition d containing N_d data instances is labeled with a vector $y_d \in \mathbb{N}^{N_d \times 1}$ encoding the allocation of data instances to partition elements.

Our learning algorithms require human labeled partitions but not the underlying links responsible for generating the partitions. The mapping from links to partitions is many-to-one and exponentially many link combinations exist that can generate an observed partition. Labeling links would involve enumerating over this exponentially large set and is clearly infeasible. Instead, we develop algorithms that estimate the probability of a link between two data instances (or latent clusters) without directly observing training links. This is a significantly more involved

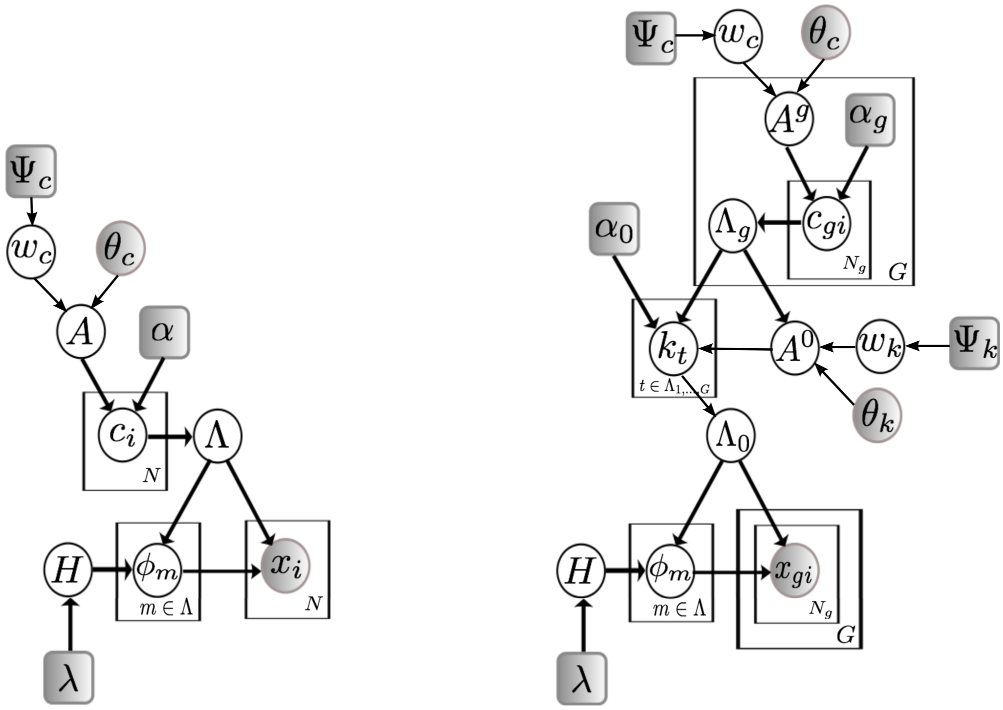


FIGURE 6.1: Covariate augmented ddCRP and hddCRP. The affinities are modeled via a weighted combination of the covariates.

problem than that has been previously addressed in the literature. Affinity learning work [135–138] that aims to estimate the pairwise probability of data instances belonging to a common partition component from observed partitions, is perhaps closest to our work. However, since there are no latent links to reason about, learning is simpler and off the shelf tools like logistic regression [135, 136] have proved effective.

Our algorithms that approximately marginalize the exponentially large set of latent links and learn the marginal distribution $p(w \mid Y)$. The corresponding joint distribution $p(w, Y) = p(w)p(Y \mid w)$ requires the specification of the likelihood model $p(Y \mid w)$. As we have seen (Figure (1.3)), human interpretations of images and videos vary wildly. Designing likelihoods that model the noise process responsible for producing different human partitions of an image is challenging. We bypass this issue by resorting to recent advances in likelihood free approximate Bayesian computation(ABC) [139]. ABC algorithms assume that it is possible to simulate data from a simulation of the likelihood model, even though the likelihood itself might be intractable. Inferences about latent variables are then made by matching *summary* statistics of the simulated and observed data. Various ABC algorithms have been proposed in the literature, here we consider MCMC based ABC algorithms which are known to sample from a target distribution restricted to some neighborhood around the observed data [139]. In the context of learning from partitions, this requires the ddCRP and hddCRP models to concentrate their probability mass on partitions "similar" to human produced partitions. We thus have the following model of human like partitions:

$$p(\mathbf{c}, \mathbf{k}, w, Y) \propto p(w) \prod_{d=1}^{D} p(\mathbf{c}_d \mid w_c) p(\mathbf{k}_d \mid \mathbf{c}_d, w_k) \delta(z(\mathbf{c}_d, \mathbf{k}_d), y_d), \tag{6.2}$$

$$\delta(y_a, y_b) = \begin{cases} 1 & \text{if } \Delta(y_a, y_b) < \epsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{6.3}$$

where for notational simplicity, we have dropped the explicit dependence on the various hyperparameters. The model restricts it's probability mass over partitions to those observations that are at most ϵ away from the ground truth partition. The notion of closeness is modeled via a loss function $\Delta(y_a, y_b)$. With an appropriate loss function and threshold ϵ the marginal posterior density $p(w \mid Y)$ will concentrate around realizations of w that favor human annotated partitions. We can then summarize the marginal posterior via its MAP estimate $\hat{w} = \underset{w}{\text{argmax}}\, p(w \mid Y)$. More sophisticated estimates that better account for the posterior uncertainty are certainly possible and constitute planned future work. For the tasks of image and video segmentations considered in this paper, we use a loss function based on the Rand index [45].

$$\Delta(y_a, y_b) = 1 - \text{RI}(y_a, y_b), \tag{6.4}$$

Observe that the likelihood in Equation 6.2 is only specified to within a constant of proportionality. This is because normalizing the loss aware likelihood $\delta(y_a, y_b)$ involves computing a summation over an exponentially large set. Importantly, our algorithms do not require the evaluation of this normalization constant.

MCMC-ABC [139, Algo. 3] algorithms are typically initialized via a rejection sampler that samples the prior distribution till a sample within the threshold is encountered. Such an initialization procedure is extremely inefficient in the high dimensional space of partitions, and would render the entire algorithm ineffective. Instead, we initialize our samplers with a human annotated partition, thus bypassing the need for rejection sampling.

Learning ddCRP weights We first consider the covariate dependent ddCRP model. Here, Equation 6.2 simplifies to

$$p(\mathbf{c}, w_c, Y) \propto p(w_c) \prod_{d=1}^{D} p(\mathbf{c}_d \mid w_c) \delta(z(\mathbf{c}_d), y_d), \tag{6.5}$$

We explore the posterior $p(\mathbf{c}, w_c \mid Y)$ by embedding a random walk Metropolis Hastings step within the ddCRP Gibbs sampler. We proceed by proposing w_c from a Gaussian distribution:

$$w_c^* \sim \mathcal{N}(w_c, \nu \mathbf{I}), \tag{6.6}$$

where ν is a free parameter controlling the scale of the proposals. The proposed w_c^* is accepted with probability $\propto \min(1, \rho_c)$ where ρ_c is:

$$\rho_c = \frac{p(\mathbf{c}, w_c^*, Y) q(w_c \mid w_c^*)}{p(\mathbf{c}, w_c, Y) q(w_c^* \mid w_c)} = \frac{p(w_c^*) \prod_d p(\mathbf{c}_d \mid w_c^*)}{p(w_c) \prod_d p(\mathbf{c}_d \mid w_c)} . \tag{6.7}$$

Next, we sample cluster links $\mathbf{c}$ using a Gibbs step:

$$\begin{aligned} c_{di} \mid \mathbf{c}_{-di}, w_c, Y &\sim p(c_{di} \mid \mathbf{c}_{-di}, w_c, Y) \\ &\sim p(c_{di} \mid w_c) \delta(z(\mathbf{c}_d), y_d). \end{aligned} \tag{6.8}$$

Neither sampling step involves evaluating the likelihood's normalization constant. After running the sampler for a sufficiently long period of time and collecting S samples, we can estimate the MAP sample $\hat{w}$,

$$\hat{w} \approx \underset{w \in \{w^{(1)}, \ldots, w^{(S)}\}}{\operatorname{argmax}} \sum_{s'=1}^{S} p(\mathbf{c}_d^{(s')} \mid w) p(w, Y) . \tag{6.9}$$

Learning hddCRP weights Learning in the hddCRP involves exploring the posterior of Equation 6.2. We proceed analogously to the ddCRP case by proposing

w from a random walk Gaussian proposal and accept it with probability proportional to $\min(1, \rho_k)$, where ρ_k is given by:

$$\rho_k = \frac{p(\mathbf{c}, w_c^*, Y) q(w_c, w_k \mid w_c^*, w_k^*)}{p(\mathbf{c}, w_c, Y) q(w_c^*, w_k^* \mid w_c, w_k)}$$

$$= \frac{p(w_c^*) \prod_d p(\mathbf{c}_d \mid w_c^*) p(\mathbf{k}_d \mid \mathbf{c}_d, w_k^*)}{p(w_c) \prod_d p(\mathbf{c}_d \mid w_c) p(\mathbf{k}_d \mid \mathbf{c}_d, w_k)} \ . \tag{6.10}$$

Conditioned on w^*, we sample the links $\mathbf{c}, \mathbf{k}$ using the algorithm presented in Section 5.2. We again collect S samples and estimate the MAP sample by marginalizing over the link variables.

$$\hat{w} \approx \underset{w \in \{w^{(1)}, \ldots, w^{(S)}\}}{\operatorname{argmax}} \sum_{s'=1}^{S} p(\mathbf{c}_d^{(s')}, \mathbf{k}_d^{(s')} \mid w) p(w, Y) \ . \tag{6.11}$$

6.3 Applications

In this section we explore its properties as well as the learning and inference algorithms developed in the previous sections. We then present results on the tasks of image, video, activity and discourse segmentation.

6.3.1 Image Segmentation

Image segmentation is the problem of partitioning an image into self-similar groups of adjacent pixels. Segmentation is an important step towards other tasks in image understanding, such as object recognition, detection,or tracking. We model images as observed collections of "superpixels" [140], which are small blocks of spatially adjacent pixels. Given a collection of superpixels our aim is to find segments made up of superpixels homogeneous in appearance *and* whose size statistics loosely match with human annotated segments. Further, we restrict ourselves to the problem of single image segmentation with $G = 1$ and drop the explicit dependence on g from our notation.

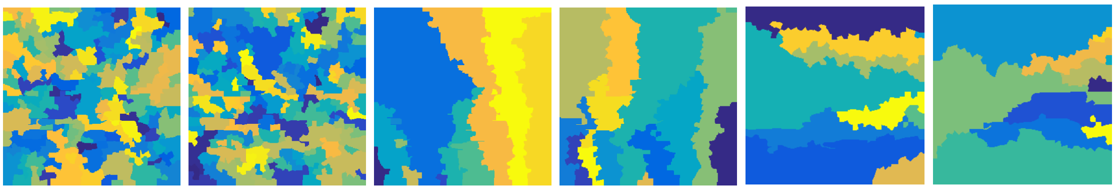


FIGURE 6.2: Partitions sampled from ddCRPs using various fixed affinity functions. The first two samples are sampled from a ddCRP using a fixed symmetric affinity function – $A_{ij} = (1 - b_{ij}) \times \mathbf{1}[i, j]$, the final four samples utilize asymmetric affinites – $A_{ij} = (1 - b_{ij}) \times \mathbf{1}[i, j] \times \mathbf{1}[(y_i - y_j) \geq 0]$ and $A_{ij} = (1 - b_{ij}) \times \mathbf{1}[i, j] \times \mathbf{1}[(r_i - r_j) \geq 0]$.

We use image segmentation to explore the benefits of learning ddCRP affinities over using manually specified affinities. In particular, we compare the covariate dependent ddCRP against the fixed affinity ddCRP on standard image segmentation benchmarks, to empirically quantify the effects of learning.

6.3.1.1 Data

We benchmark the models on two image segmentation datasets. The first dataset comprises a collection of images drawn from eight natural scene categories [141] available as a subset of the LabelMe [142] dataset.[1] The images come annotated with human segmentations, performed by non-expert users. For each category we select 150 images using the first 50 for training and the rest for testing. We also benchmark performance on the Berkeley image segmentation dataset (BSDS300) [143] using the standard train and test splits. The segmentations produced by the competing methods are quantitatively evaluated with respect to human segmentations via the Rand index [45].

As a preprocessing step, we divide each image from the two datasets into approximately 1000 superpixels [140, 144] [2] using the normalized cut algorithm [145].[3]

[1] *labelme.csail.mit.edu/browseLabelMe/*

[2] *www.cs.sfu.ca/~mori/*

[3] *www.eecs.berkeley.edu/Research/Projects/CS/vision/*

6.3.1.2 Prior

We consider a few different ddCRP priors. First, for the fixed affinity version (*ddCRP*) we manually specify data affinities that encourage spatial neighbors not separated by strong intervening contours to connect to one another by setting $A_{ij} = (1 - b_{ij}) \times \mathbf{1}[i, j]$. Here, $0 \leq b_{ij} \leq 1$ is the maximum Pb [26] response along a straight line segment connecting the centers of superpixels i, j, and $\mathbf{1}[i, j]$ takes a value of 1 if i and j are spatial neighbors, and 0 otherwise. The self connection parameter α is set to 10^{-8}. The restriction to immediate spatial neighbors guarantees spatially connected segments, a desirable property for image segmentation algorithms. However, samples from such priors exhibit severe over segmentation (Figure 6.2). This is caused by the symmetric nature of spatial affinities between superpixels. It allows neighboring superpixels i and j to link to one another with equal probability and gives rise to several small groups of interconnected superpixels. To tackle the over segmentation problem [52] used the naive-hddCRP to specify a prior over image segmentations. Alternatively, one could use the ddCRP over an ordered collection of superpixels to break the symmetry exhibited by the spatial affinity function. Figure 6.2 illustrates samples from two such affinity functions. The first, allows links only between a superpixel and its immediate neighbors to the left and the second restricts a superpixel's connections to immediate neighbors to the north giving rise to segmentations that favor horizontal and vertical structures.

Next, instead of relying on the user to determine the appropriate superpixel ordering we specify covariates, signed distances between superpixel locations along x and y axes ($\delta x = r_i - r_j$, $\delta y = y_i - y_j$), for encoding superpixel orderings. Here, r_i and y_i represent the x and y location of superpixel i. The relative importance of these structural covariates are learned from data. Together with b_{ij} they specify the *learned-ddCRP* prior over image partitions.

$$\begin{aligned} A_{ij} &= f(w, i, j) = (1 + exp(d_{ij}))^{-1} \times \mathbf{1}[i, j], \\ d_{ij} &= w_c^T \theta_{ij}^c = w_c^T [\frac{r_i - r_j}{R}, \frac{y_i - y_j}{Y}, b_{ij}]^T, \end{aligned} \tag{6.12}$$

where $R = \max(|r_i - r_j|)$ and $Y = \max(|y_i - y_j|)$.

Both ddCRP and learned-ddCRP, through their dependence on image contours, describe conditional priors on image partitions. We also consider a generative version that only considers superpixel locations: $\theta^c_{ij} = w_c^T [\frac{r_i - r_j}{R}, \frac{y_i - y_j}{Y}]^T$.

Qualitative comparisons Figure 6.3 illustrate partitions sampled from the learned ddCRP. We consider both generative and conditional affinities. The generative affinities learn more general characteristics of the scene category, for instance the tall buildings category contains partitions with vertical structures while the mountain category consists of more triangular structures. Conditional samples adapt to particular images and more closely reflect the particular structure of the image being conditioned on.

Figure 6.4 presents summary statistics computed from $10,000$ partitions sampled from learned generative affinities. We find that the Forest, Street and Inside city categories on average have a larger number of segments per partition. The ground truth partitions of these categories contain a large number of small segments, as a result we learn weights that prefer smaller segments. In contrast, the Coast and Highway category human partitions contain fewer but larger segments. This is again reflected in the learned weights, partitions of these categories contain fewer segments. We also find that the segment sizes in the learned partitions roughly follow a power law distribution, across all categories. This is a well known property exhibited by natural image segmentations [11].

6.3.1.3 Likelihood

We describe the texture of each superpixel via a local texton histogram [146], using band-pass filter responses quantized to 128 bins. A 120-bin HSV color histogram is used to describe the color of the superpixel. Each superpixel i is summarized via these histograms $x_i = \{x_i^c, x_i^t\}$. These histograms are treated as conditionally independent given the cluster allocations z and are modeled as samples from multinomial distributions with Dirichlet priors.

$$\begin{aligned} x_i^c &\sim \text{Mult}(\phi^c_{z_i}), \qquad \phi^c_{z_i} \sim \text{Dir}(\lambda^c), \\ x_i^t &\sim \text{Mult}(\phi^t_{z_i}), \qquad \phi^t_{z_i} \sim \text{Dir}(\lambda^t). \end{aligned} \tag{6.13}$$

Hyperparameters The multinomial likelihoods treat pixels within a superpixel as independent random variables. However, the ddCRP prior models affinities between superpixels. This can cause the prior to get washed away in favor of the likelihoods. To rectify this we introduce a hyperparameter γ that controls the relative importance of the prior and the likelihood,

$$p(\mathbf{x}, \mathbf{c} \mid \alpha, A, \gamma, \lambda) \propto p(\mathbf{c} \mid \alpha, A) \ \ \{p(\mathbf{x} \mid \mathbf{c}, \lambda)\}^{\gamma}. \tag{6.14}$$

The Dirichlet hyperparmaeters $\lambda = \{\lambda_c, \lambda_t\}$ along with γ are learned via a grid search on the training set. Given a grid of possible hyperparameters we hill climb on the posterior probability surface by running a small number of MCMC iterations. Finally, we select the set of hyperparameters that produce optimal results according to a chosen loss function, Rand index in this case. For the Dirichlet hyper-parameters we searched over a coarse grid located at locations:
$\{0.01, 0.1, 1, 5, 10, 20, 25, 40, 50, 100\}$, for γ we searched over the range:
$\{0.001, 0.005, 0.05, 0.01, 0.1, 1, 10\}$.

6.3.1.4 Empirical Comparisons

In addition to the various ddCRP models, we also compare to a image segmentation algorithm based on the gPb boundary detector that achieves state-of-the-art results [20] on standard benchmarks. It has one tunable scale parameter which we tune on the training sets. We learn independent models and also search the optimal gPb scale for each of the eight LabelMe categories independently. We ran 500 iterations of the MCMC samplers for the ddCRP variants and selected the MAP sample as the desired segmentation. The performance summary is presented in Figure 6.4 and qualitative comparisons can be found in Figure 6.5. Armed with well tuned likelihoods all three ddCRP models perform well. Nonetheless, learning the affinities provides a modest but statistically significant gain. Of the eight categories, the learned-ddCRP outperforms ddCRP on two categories (Outside and Tall Buildings) and is statistically indistinguishable on the remaining six. The learned ddCRP using only the impoverished generative features also manages to be competitive with the conditional ddCRP model. It outperforms ddCRP on images from the Street category while being worse on Highway images. Image contours are a strong cue for segmentation utilized by ddCRP but not by the generative version. The generative version being competitive inspite of this can be attributed to learning.

The gPb based algorithm is outperformed by the learned-ddCRP on five categories, worse on one and within noise on the rest. The contour responses on LabelMe images is weaker causing gPb to not perform as well.

6.3.2 Video Segmentation

Finally, we consider the problem of discovering segments from videos that are coherent in space, time and appearance. The problem is a natural fit for the hierarchical ddCRP. We model video frames using independent spatial ddCRPs and couple them using a temporal ddCRP. As with image segmentation, instead of working with pixels we preprocess the video into a collection of superpixels.

6.3.2.1 Data

We perform experiments on of the recently introduced VSB100 [67] dataset. It contains 40 training and 60 testing videos. We restrict our attention to the general benchmark subset where the 100 videos are annotated by multiple human subjects with temporally smooth segments coherent in appearance.

6.3.2.2 Prior

The hddCRP prior requires affinity functions to be specified between both data instances and clusters. We experiment with both learned and manually specified affinity functions. In the learned case (*learned-hddcrp*), we reuse the image segmentation affinity functions between data instances. Affinity between clusters t, s is expressed as a linear weighted combination of covariates (θ^k_{ts}) encoding shape, size and positional affinities,

$$\theta^k_{ts} = [\vartheta_{ts}, \varphi_{ts}, \frac{|\zeta_t - \zeta_s|}{S}]^T. \tag{6.15}$$

The variable ζ_t denotes the size of cluster t and $S = \max|\zeta_t - \zeta_s|$. The covariates collectively represented by ϑ_{ts} capture within frame affinities and are defined as follows:

$$\vartheta_{ts} = \mathbf{1}_{[t,s|t \in g, s \in g]} \big[\frac{r_t - r_s}{R}, \frac{y_t - y_s}{Y}\big]^T. \tag{6.16}$$

Across frame affinities are captured in φ_{ts},

$$\varphi_{ts} = \mathbf{1}_{[t,s|t \in g+1, s \in g]} \big[\frac{|r_t - r_s|}{R}, \frac{|y_t - y_s|}{Y}, 1 - \frac{t \cap s}{t \cup s}\big]^T. \tag{6.17}$$

Within a frame we capture similarity between cluster locations using signed Manhattan distances. Across frame positional similarities are captured using standard Manhattan distances and through an intersection over union measure of the projection of one cluster on another. Finally, the affinity between clusters t, s is modeled via a sigmoidal transformation:

$$\begin{aligned} d_{ts} &= w_k^T \theta_{ts}^k, \\ A_{ts}^0 &= (1 + exp(d_{ts}))^{-1}. \end{aligned} \tag{6.18}$$

We also consider a version of the *naive-hddCRP* (Section5.1.2) that employs covariate dependent affinity functions at the data level but at the cluster level resorts to CRP affinities.

6.3.2.3 Likelihood

As a preprocessing step, we divide each frame into approximately 1200 superpixels using the method proposed by Chang et al. [147].[4] Following the image segmentation likelihood model, we describe a superpixel using 120-bin HSV color and 128-bin local texton histograms. The color and texture features for super-pixel i in video frame g are denoted by $x_{gi} = \{x_{gi}^c, x_{gi}^t\}$, where

$$\begin{aligned} x_{gi}^c &\sim \text{Mult}(\phi_{z_{gi}}^c), \phi_{z_{gi}}^c \sim \text{Dir}(\lambda^c), \\ x_{gi}^t &\sim \text{Mult}(\phi_{z_{gi}}^t), \phi_{z_{gi}}^t \sim \text{Dir}(\lambda^t). \end{aligned} \tag{6.19}$$

[4] Chang et al. [147] also estimate temporal correspondences between superpixels, but we do not utilize this information.

The proposed likelihood model forces clusters across video frames belonging to the same video segment share a common appearance model, encoding the assumption that appearance of objets doesn't change significantly over the course of the video. More elaborate likelihoods could be developed to capture appearance changes and is interesting future work.

As with image segmentation, in addition to the Dirichlet hyperparameters controlling the texture and color likelihoods we introduce an additional parameter controlling the relative importance of the likelihood,

$$p(\mathbf{x}, \mathbf{k}, \mathbf{c} \mid \alpha_{1:G}, \alpha_0, A^{1:G}, A^0, \lambda) \propto p(\mathbf{c}, \mathbf{k} \mid \alpha_{1:G}, \alpha_0, A^{1:G}, A^0) \ \ \{p(\mathbf{x} \mid \mathbf{c}, \mathbf{k}, \lambda)\}^{\gamma}. \tag{6.20}$$

All likelihood hyperparameters are learned through validation analogously to image segmentation.

6.3.2.4 Empirical Comparisons

We quantify video segmentation performance using two measures: probabilistic Rand index (PRI) and volumetric precision and recall (VPR) [67]. In order to penalize spatially coherent but temporally inaccurate segmentations that exhibit frequent "label switching" between video frames we compute the segmentation quality measures by treating the entire video sequence as a single spatio-temporal block.

We compare video segmentation performance against two state-of-the-art video segmentation algorithms. First, we consider the latest iteration[5] of a popular non probabilistic *hierarchical graph-based video segmentation* (HGVS) algorithm [53]. We also compare against the algorithm (VSS) proposed in [148] which has been shown to perform well on the VSB100 dataset. For both these methods we present the numbers reported in [67]. For the hddCRP variants, we ran three MCMC chains each for 1000 iterations and selected the MAP video segmentation. Figure 6.8 provides a qualitative summary of this experiment. In spite of using identical likelihood models, the learned-hddCRP consistently produces segments that are visually cleaner, exhibit smaller under segmentation errors and better temporal

[5] http://www.videosegmentation.com/

coherence. These qualitative results translate to improved empirical performance (Figure 6.8), with leanred-hddCRP outperforming the naive version both in terms of PRI and VPR. Together these results demonstrate the effectiveness of learning the hddCRP affinities over manually specified affinities. The learned-hddCRP performs comparably to both the VSS and the HGVS and is statistically indistinguishable from either.

Limitations A glance at Figure 6.8 reveals that naive-hddCRP and to a lesser degree the learned-hddCRP tend to under segment videos. This preference can be attributed to the biases induced by the likelihood model. First, multinomial likelihoods with high probability allow histograms of widely varying shapes and thus segments that merge regions with distinct appearance are often preferred, especially when the Dirichlet hyper-parameters are set to large values. Next, since a common likelihood model is used to describe the entire video we are unable to model subtle appearance variations across time. Thus, large Dirichlet hyperparameters are required to explain temporally consistent video segments that span multiple video frames in the validation set. The validated hyper-parameters thus tend to be large and give rise to the observed undersegmentation errors. Interesting future work would involve exploring different likelihood models that use alternate exponential family distributions while decoupling temporal and spatial consistency.

6.4 Discussion

In this chapter, we dealt with the difficult problem of designing ddCRP and hddCRP affinity functions. We developed algorithms for automatically learning effective affinity functions from human annotated clusterings. The learned affinity functions provided clear, demonstrable benefits of over manually crafted counterparts, popular in the literature. On the tasks of image and video segmentation our learned models performed competitively with established image and video segmentation algorithms.

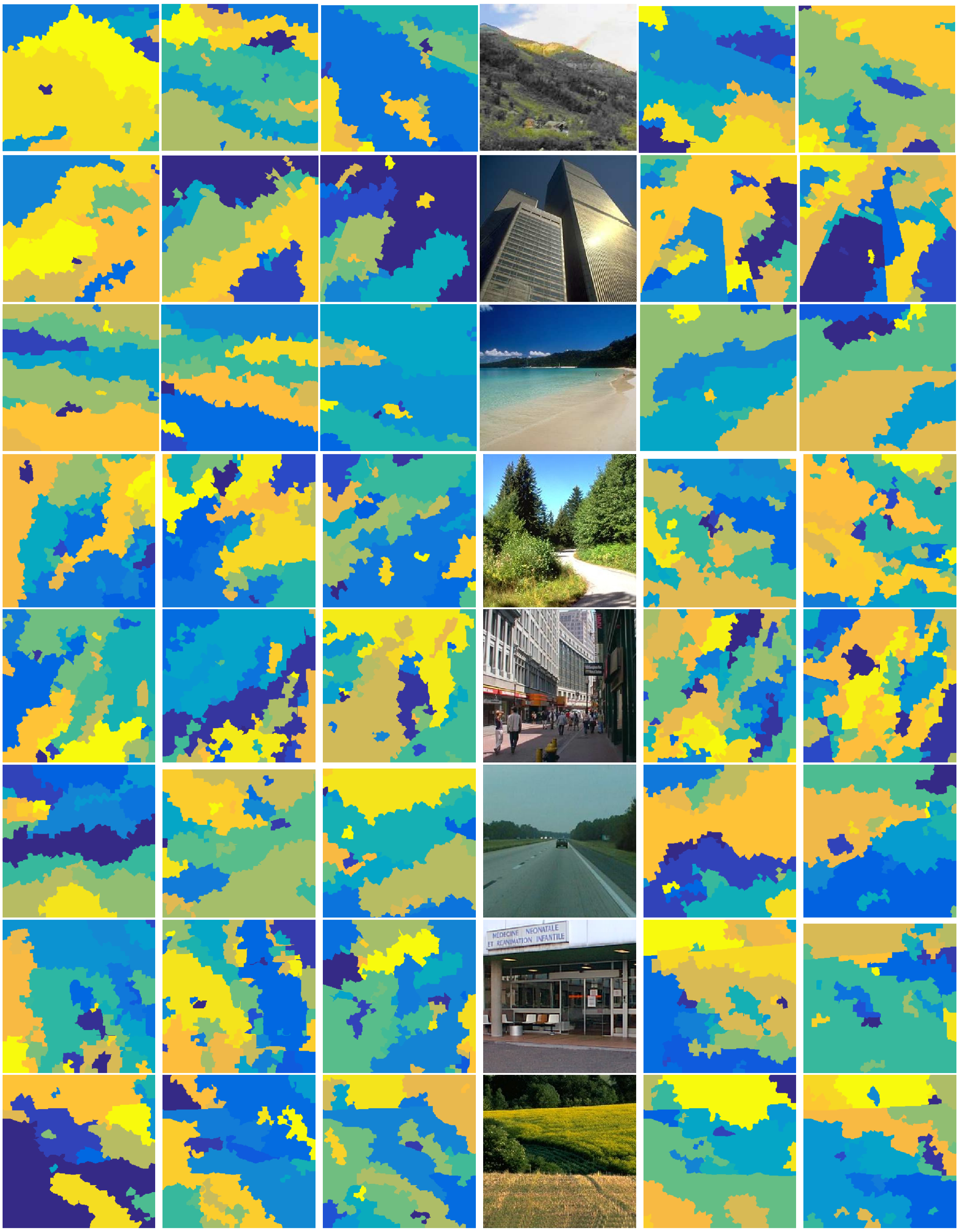
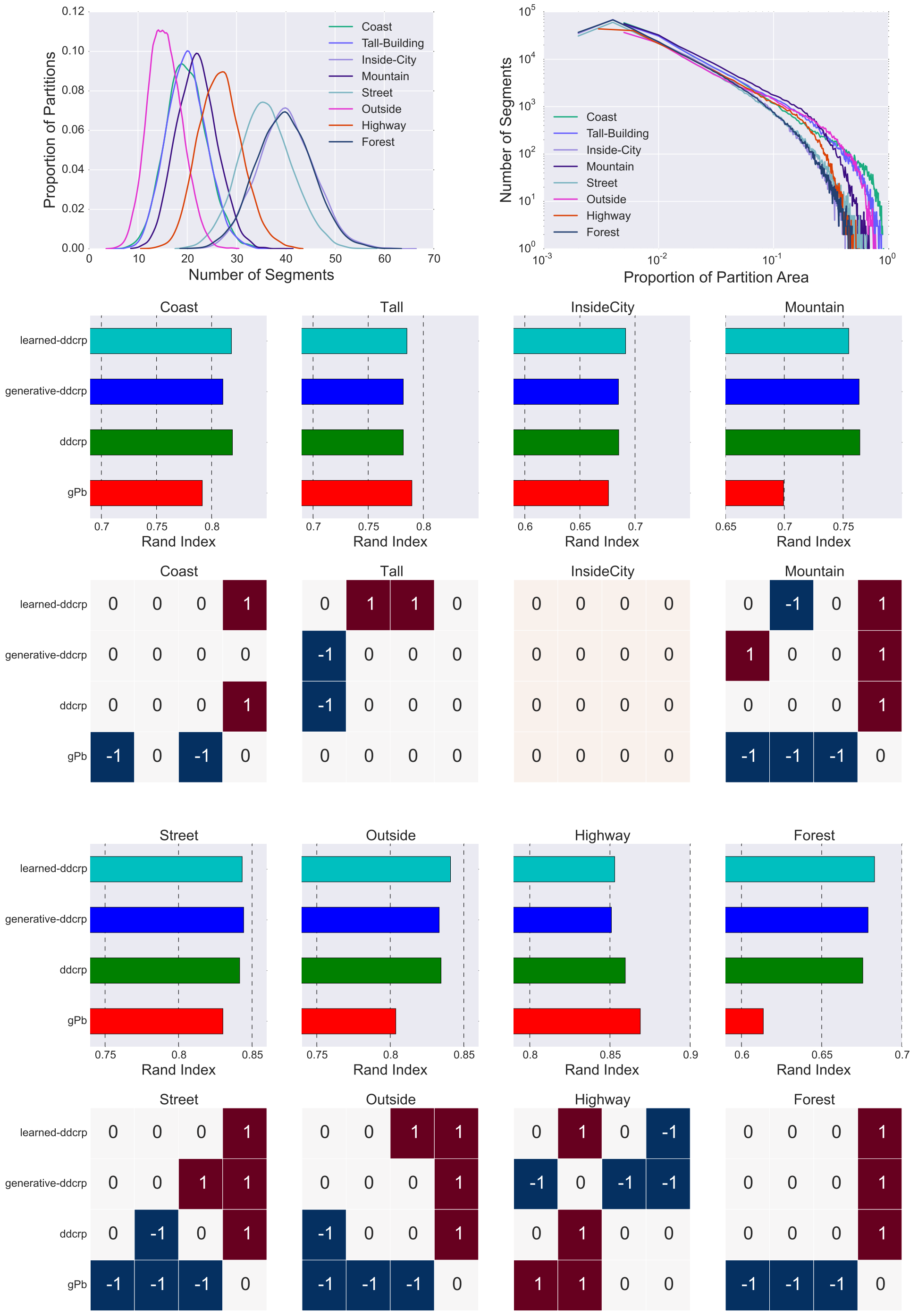


FIGURE 6.3: Samples from ddCRP **priors** with learned affinities. Rows display samples from a ddCRP model trained on the Mountain, Tall building, Coast and Forest categories. The first three columns correspond to generative samples while the two rightmost columns were generated by conditioning on the displayed image.



Coast

learned-ddcrp	0	0	0	1
generative-ddcrp	0	0	0	0
ddcrp	0	0	0	1
gPb	-1	0	-1	0

Tall

learned-ddcrp	0	1	1	0
generative-ddcrp	-1	0	0	0
ddcrp	-1	0	0	0
gPb	0	0	0	0

InsideCity

learned-ddcrp	0	0	0	0
generative-ddcrp	0	0	0	0
ddcrp	0	0	0	0
gPb	0	0	0	0

Mountain

learned-ddcrp	0	-1	0	1
generative-ddcrp	1	0	0	1
ddcrp	0	0	0	1
gPb	-1	-1	-1	0

Street

learned-ddcrp	0	0	0	1
generative-ddcrp	0	0	1	1
ddcrp	0	-1	0	1
gPb	-1	-1	-1	0

Outside

learned-ddcrp	0	0	1	1
generative-ddcrp	0	0	0	1
ddcrp	-1	0	0	1
gPb	-1	-1	-1	0

Highway

learned-ddcrp	0	1	0	-1
generative-ddcrp	-1	0	-1	-1
ddcrp	0	1	0	0
gPb	1	1	0	0

Forest

learned-ddcrp	0	0	0	1
generative-ddcrp	0	0	0	1
ddcrp	0	0	0	1
gPb	-1	-1	-1	0

FIGURE 6.4: *Top.* Summary statistics of partitions sampled from ddCRP models with learned generative weights. *Left:* Empirical distribution of the number of segments, broken down by the eight natural image category. *Right:* Number of segments occupying varying proportions of the image area, on a log-log scale. *Bottom.* Segmentation performance on the eight LabelMe categories. Rows 1 and 3 display Rand index (higher is better) achieved by competing models. Rows 2 and 4 present results from a Wilcoxon's signed rank test. Statistically indistinguishable, better and worse methods are denoted by 0, 1 and -1 respectively at 95% confidence interval.

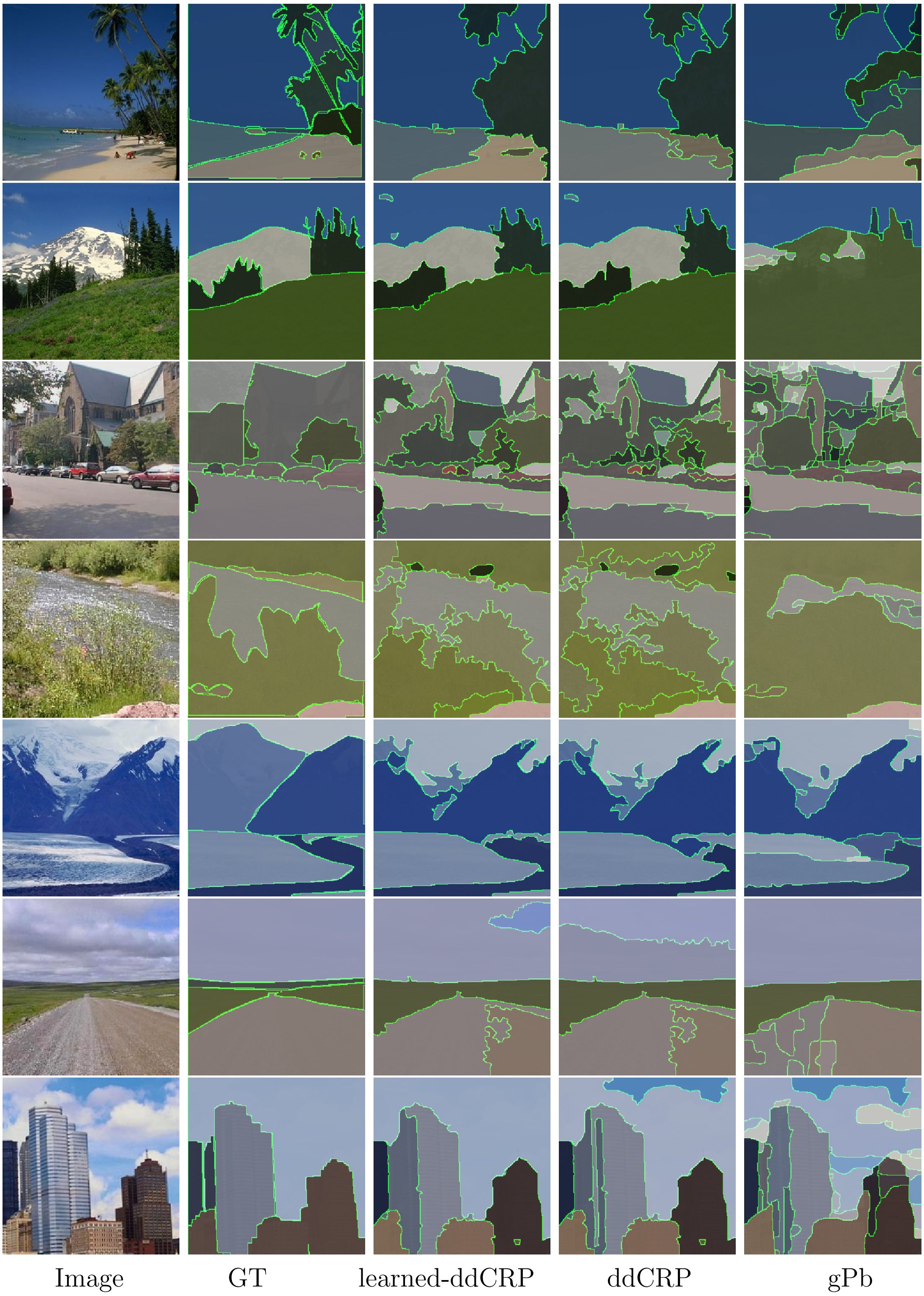


Figure 6.5: Rand index produced by competing methods on the LabelMe dataset. From left to right we have, the original image, the human segmentation, segmentations produced by learned ddCRP, naive ddCRP and gPb.

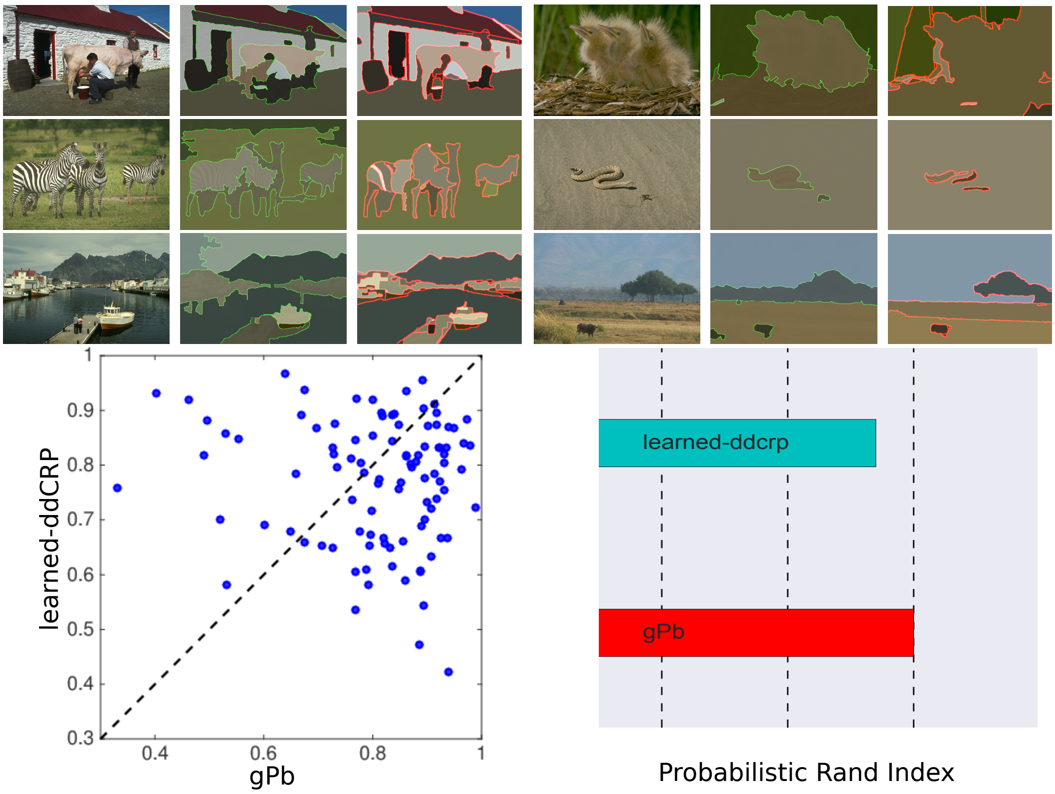


Figure 6.6: Segmentations produced by learned-ddCRP and gPb on BSDS. *Bottom-right*: Performance in terms of probabilistic Rand index.

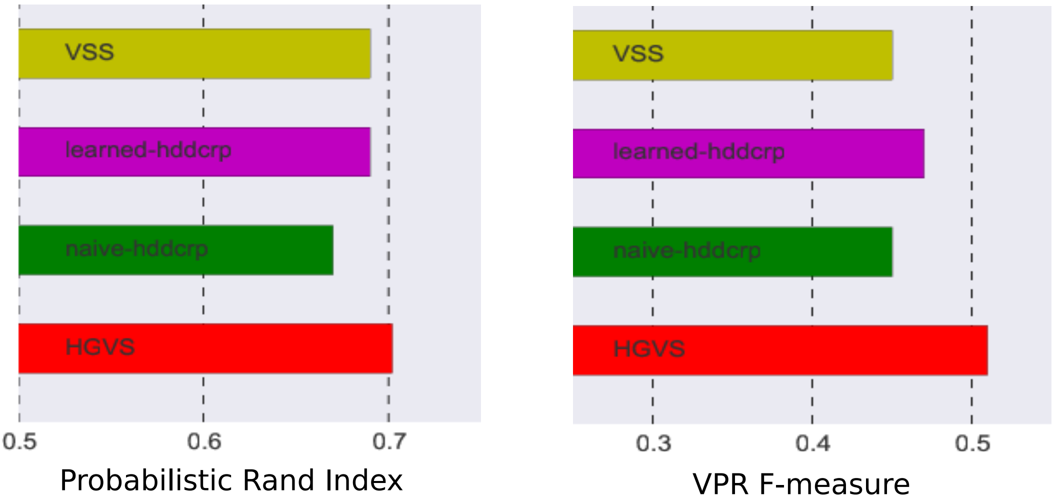


Figure 6.7: Quantitative performance on VSB100 video segmentation benchmark. Probabilistic Rand index on the left and volumetric precision recall on the right.

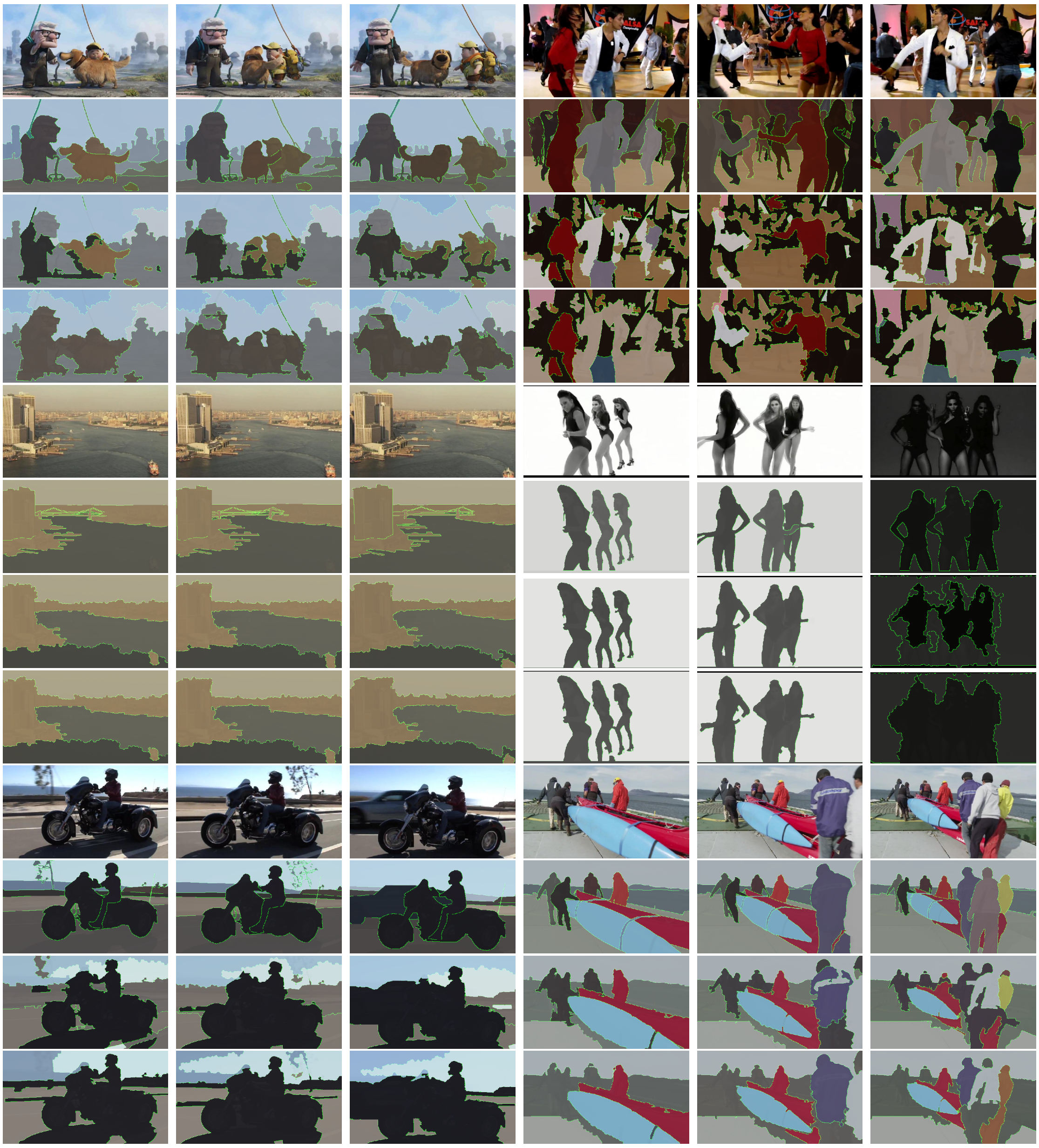


Figure 6.8: Examples from VSB100 test set. For each video the first, middle and last frames are displayed. The row immediately below the video displays the ground truth. The following two rows display segmentations produced by learned and naive-hddCRP models.

Chapter 7

Contributions and Recommendations

In the preceding chapters, we have introduced statistical models and methods for discovering layers from images, segments from videos, parts from 3D representation of objects, activities from MoCap data and discourse units from collections of related documents. Here, we summarize our main contributions and discuss interesting directions of future research.

7.1 Summary of Contributions

Discovery of regions, parts, activities and discourse units from scenes, objects, sensor streams and documents are all examples of ill-posed problems. To make progress, assumptions about underlying physical processes are necessary. In this thesis, we have focused on developing flexible statistical priors that attempt to closely model physical regularities while lending themselves to the development of efficient computational algorithms.

In Chapter 3, we extend the layered segmentation model of Sudderth and Jordan [22] to more closely match the statistics of natural image segmentations. We develop new algorithms for learning conditional versions of the model from large collections of natural image segmentations. We find that such conditionally specified models more closely match the statistics of human image segmentations and lead to improved segmentation performance.

Performing effective inference in the layered model is challenging. The mean field approach suggested in [22] is fraught with local optima issues and leads to unreliable results. We develop a sophisticated discrete optimization based inference algorithm that escapes local optima through large moves in the partition space. The algorithm requires a marginalization over an infinite set of thresholded Gaussian processes, a challenging undertaking. To address this, we resort to an expectation propagation based message passing algorithm for approximately (but accurately) performing such marginalizations. Through extensive experiments, we find that the learning and inference enhancements result in significant performance improvements producing results competitive with *state-of-the-art* image segmentation techniques.

Chapter 4 develops methods for discovering parts from observed articulations of deformable objects. To the best of our knowledge, the model developed in Chapter 4 is the first to simultaneously infer the number and spatial extent of parts while guaranteeing that the inferred parts are spatially contiguous. We adapt the distance dependent Chinese restaurant process prior to define a distribution over partitions of objects that places zero probability mass on partitions with spatially non-contiguous parts. We model the observed affine deformations through a matrix variate normal distribution. Studies on a large corpus of human body scans of widely varying shapes and poses demonstrate the effectiveness of our methods.

Hierarchical extensions to the ddCRP are developed in Chapter 5. Such hierarchical models allow us to perform shared segmentation of groups of related data. Hierarchical ddCRP (hddCRP) models non-exchangeability both between data instances inside a group and between clusters of data items across groups. Through the specification of affinities between data points and between clusters, data exhibiting very different structures can be easily modeled. Inference in such models is challenging and necessitates the development of new Metropolis Hastings based algorithms that make coordinated changes at both data and cluster levels. We demonstrate the effectiveness of the hddCRP models by applying them to the diverse tasks of activity recognition and discourse segmentation. On both tasks, we find that the hddCRP achieves *state-of-the-art* performance.

Our final contribution, involves developing algorithms for automatically learning effective affinity functions from human annotated data partitions (Chapter 6). Borrowing ideas from recent advances in approximate Bayesian computation (ABC), we show that it is possible to learn affinity functions between data instance and their clusters from human clusterings without needing to observe the links responsible for generating the clustering. We find that such learned affinity functions lead to a significant performance boost over hand crafted affinities and extremely competitive results on standard image and video segmentation benchmarks.

7.2 Recommendations

We conclude with a discussion of the limitations of the proposed methods and exciting avenues of research to remedy them.

7.2.1 Image and Video Segmentation

Improved Likelihood Models While a significant focus of this thesis has been on developing realistic priors on partitions of natural images and videos, our treatment of region and object appearances have been quite rudimentary. We have modeled quantized color and texture responses via multinomial likelihoods. While such likelihoods are popular in the literature, they are primarily motivated by computational ease rather than modeling capacity. Such multinomial likelihoods ignore correlations across bins, incorrectly treat pixels within a super pixel as independent observations and depending on hyper parameter settings do not object to grouping together pixels with vastly different appearances. Covariance based region descriptors [149] capture feature correlations and allay many of the issues with multinomial likelihoods. Wishart and inverse-Wishart distributions have support over positive definite matrices and are natural choices for modeling region covariance descriptors. Relatively recent work [150] has used such likelihoods for clustering faces and matching image patches across video frames but application to segmentation problems appears to be as yet unexplored. Nonetheless, combining such likelihoods with our sophisticated segmentation priors will likely lead to improved performance.

Image Understanding Systems Recent years have seen significant progress in object recognition and image labeling [3, 4]. State-of-the-art systems place bounding boxes around objects in images and optionally produce a dense labeling into one of K predefined classes. Such labelings while useful only provide a superficial understanding of the image. Humans on the other hand, are additionally able to infer rich geometric structure and disambiguate occlusion and support relationships by reasoning about the underlying 3D scene space rather than the 2D image space. The inability to reason in 3D is one plausible explanation for computer vision systems falling well short of human performance on image understanding tasks. Coherent statistical models of objects, their appearances in 2D and their shapes and poses in the encompassing 3D scene provide an exciting direction for bridging this gap. The layered representation introduced in Chapter 3 is a promising building block for such models. Extending the latent layers with explicit depth and shape parameters will allow coarse modeling of the underlying 3D scene responsible for generating an image or a collection of images. Further, the layered representation could be shared among images allowing for the sharing of statistical strength and better characterize ambiguous image regions. Sharing latent layered representations among related images instead of observed appearance features provides robustness to appearance variations arising from occlusion effects and changes in viewpoint. Development of such models and corresponding computational algorithms will likely lead to better image understanding systems.

7.2.2 Articulated object segmentation

Our work on part discovery from observed articulations requires that the correspondence between different poses are known. Aligning large collections of 3D meshes exhibiting a diversity of poses, resolutions and shapes is a challenging problem. We currently use a two step procedure – aligning meshes through off the shelf alignments algorithms before segmenting the aligned meshes. Consequently, we are unable to recover from alignment errors.

An interesting direction for future research would involve exploring enhanced models that jointly infer correspondences between meshes and segment meshes into parts. Recent work [65], in the context of optical flow estimation, has found that models that simultaneously discover image segments and image motion lead to improvements in both segmentation and motion estimation. It is likely that similar gains can be had in the analysis of large collections of meshes when jointly tackling the correspondence and segmentation problems.

7.2.3 Scalable and Reliable Inference

The ddCRP and its hierarchical variants developed in this thesis have relied on MCMC algorithms for inference. Such MCMC methods provide strong asymptotic guarantees and can be applied to complex models relatively easily. However, for large problems it is often not possible to run MCMC chains to convergence. Furthermore, assessing convergence is itself difficult. Recent progress combining ideas from variational inference and stochastic optimization provides a promising alternative for large data collections. In the context of ddCRP, Bartunov and Vetrov [151] have recently developed variational inference schemes for ddCRP models employing sequential distances. Generalizations of these algorithms for the larger class of ddCRP and hierarchical ddCRP models may lead to more scalable inference algorithms.

The discrete stochastic search based inference algorithms developed in Chapter 3, provide another promising avenue of future research. Through explicit split and merge moves these algorithms are more robust to shallow local optima. Combining these ideas with more recent advances in combinatorial optimization will likely lead to improved inference algorithms for models with discrete latent variables.

Appendix A

Algorithmic Details from Chapter 3

A.1 Low rank Expectation Propagation

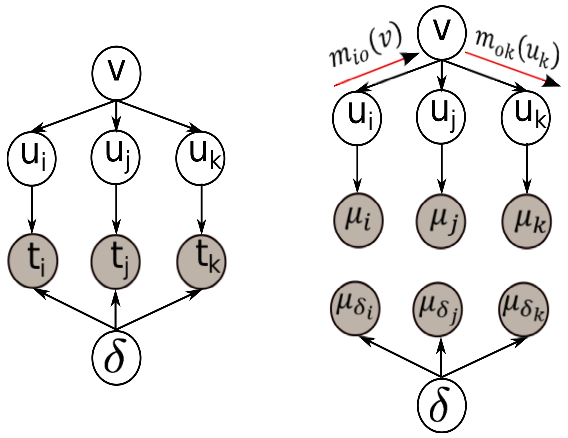


FIGURE A.1: True and Approximate distributions. Graphical models representing the distribution of random variables in a layer (*We have left out the hyper-parameters on δ and v*). **Left**: True distribution. **Right**: Approximate distribution.

As previously noted, the random variables associated with each layer of our model can be treated independently of the others. Following the notation introduced in

Section 3, we have

$$p(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) p(\delta|\alpha) \prod_{n=1}^{N} \mathcal{N}(u_n | a_n^T \mathbf{v}, \psi_n) \mathbb{I}(t_n(\delta - u_n) > 0) \tag{A.1}$$

We approximate this distribution with a Gaussian distribution of the form:

$$q(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) \mathcal{N}(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2) \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T \mathbf{v}, \psi_n) \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) \\ \mathcal{N}(\delta \mid \tilde{\mu}_{\delta_n}, \tilde{\sigma}_{\delta_n}^2) \tag{A.2}$$

The graphical models corresponding to the true and approximate distributions are shown in Figure A.1. EP proceeds by removing an approximate factor and substituting it with the corresponding true factor, giving rise to the augmented distribution. The moments of this augmented distribution are then computed and the parameters of the approximate factor is updated by matching the moments of the approximate and augmented distributions. Next, we demonstrate how these quantities are computed for our model.

Firstly, note that our approximation assumes independence between δ and $\{\mathbf{u}, \mathbf{v}\}$. From figure A.1 and using standard Gaussian BP results we have

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) \prod_{n=1}^{N} m_{no}(\mathbf{v}) \tag{A.3}$$

with

$$m_{no}(\mathbf{v}) \propto \mathcal{N}(\mathbf{v} \mid \boldsymbol{\tau}_{no}^{-1} \boldsymbol{\nu}_{no}, \boldsymbol{\tau}_{no}^{-1}), \ \boldsymbol{\tau}_{no} = \frac{\tilde{\tau}_n}{1 + \psi_n \tilde{\tau}_n} a_n a_n^T \tag{A.4}$$

$$\boldsymbol{\nu}_{no} = \frac{\tilde{\nu}_n}{1 + \psi_n \tilde{\tau}_n} a_n, \ \tilde{\nu}_n = \tilde{\tau}_n \tilde{\mu}_n, \ \tilde{\tau}_n = \tilde{\sigma}_n^{-2} \tag{A.5}$$

Thus, we have the following result

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{pos}^{-1}\boldsymbol{\nu}_{pos}, \boldsymbol{\tau}_{pos}^{-1}) \tag{A.6}$$

$$\boldsymbol{\tau}_{pos} = \boldsymbol{I} + \sum_{n=1}^{N} \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} a_n a_n^T \tag{A.7}$$

$$\boldsymbol{\nu}_{pos} = \sum_{n=1}^{N} \frac{\tilde{\nu}_n}{1+\psi_n\tilde{\tau}_n} a_n \tag{A.8}$$

We can remove the effect of an approximate factor by dividing out the corresponding message.

$$q(\mathbf{v} \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{-n}^{-1}\boldsymbol{\nu}_{-n}, \boldsymbol{\tau}_{-n}^{-1}) \tag{A.9}$$

$$\boldsymbol{\tau}_{-n}^{-1} = (\boldsymbol{\tau}_{pos} - \boldsymbol{\tau}_{no})^{-1} \tag{A.10}$$

$$\boldsymbol{\nu}_{-n} = \boldsymbol{\nu}_{pos} - \boldsymbol{\nu}_{no} \tag{A.11}$$

Note that $\boldsymbol{\tau}_{-n}^{-1}$ can be efficiently computed using the following rank one update:

$$\boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} - (-m)\frac{\boldsymbol{\Sigma} a_n a_n^T \boldsymbol{\Sigma}}{1 - m a_n^T \boldsymbol{\Sigma} a_n} \tag{A.12}$$

$$m = \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} \; \textit{and} \; \boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} \tag{A.13}$$

Next observe that

$$q(u_n|\boldsymbol{t}) \propto \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) m_{on}(u_n) \tag{A.14}$$

$$q(u_n|\boldsymbol{t}_{-n}) \propto m_{on}(u_n) \tag{A.15}$$

$$m_{on}(u_n) \propto \mathcal{N}(u_n \mid \tau_{on}^{-1}\nu_{on}, \tau_{on}^{-1}) \tag{A.16}$$

A little algebra reveals that the parameters of m_{on} are given by

$$\tau_{on}^{-1} = \psi_n + a_n^T \boldsymbol{\tau}_{-n}^{-1} a_n \; \textit{and} \; \tau_{on}^{-1}\nu_{on} = a_n^T \boldsymbol{\tau}_{-n}^{-1} \boldsymbol{\nu}_{-n} \tag{A.17}$$

Similarly, the parameters of the distribution $q(\delta \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\delta \mid \tau_{-\delta_n}^{-1}\nu_{-\delta_n}, \tau_{-\delta_n}^{-1})$ can be computed. Finally, the moments of the following augmented distribution need to be computed:

$$q(u_n, \delta \mid \boldsymbol{t}_{-n})\mathbb{I}(t_n(\delta - u_n) > 0) = q(\delta \mid \boldsymbol{t}_{-n})q(u_n \mid \boldsymbol{t}_{-n})\mathbb{I}(t_n(\delta - u_n) > 0) \quad \text{(A.18)}$$

A little bit of algebra leads to the following closed form formula for the relevant normalization constants.

Normalization constant of the augmented distribution (0^{th} order moment):

$$P = \Phi\left(\frac{t_n(\mu_{-\delta_n} - \mu_{-n})}{\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}}\right) = \Phi(h_n) \quad \text{(A.19)}$$

First and Second order moments for δ:

$$E[\delta] = \mu_{-\delta_n} + t_n \frac{\sigma_{-\delta_n}^2 \mathcal{N}(h_n)}{\Phi(h_n)\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}} \quad \text{(A.20)}$$

$$E[\delta^2] = 2\mu_{-\delta_n}E[\delta] - \mu_{-\delta_n}^2 + \sigma_{-\delta_n}^2 - \frac{\sigma_{-\delta_n}^4 h_n \mathcal{N}(h_n)}{\Phi(h_n)(\sigma_{-n}^2 + \sigma_{-\delta_n}^2)} \quad \text{(A.21)}$$

First and Second order moments for u_n:

$$E[u_n] = \mu_{-n} - t_n \frac{\sigma_{-n}^2 \mathcal{N}(h_n)}{\Phi(h_n)\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}} \quad \text{(A.22)}$$

$$E[u_n^2] = 2\mu_{-n}E[u_n] - \mu_{-n}^2 + \sigma_{-n}^2 - \frac{\sigma_{-n}^4 h_n \mathcal{N}(h_n)}{\Phi(h_n)(\sigma_{-n}^2 + \sigma_{-\delta_n}^2)} \quad \text{(A.23)}$$

where $\mu_{-n} = \tau_{on}^{-1}\nu_{on}$, $\mu_{-\delta_n} = \tau_{-\delta_n}^{-1}\nu_{-\delta_n}$, $\sigma_{-\delta_n}^2 = \tau_{-\delta_n}^{-1}$, $\sigma_{-n}^2 = \tau_{on}^{-1}$.

The parameters of the approximate factor corresponding to u_n can now be computed and the posterior on $\mathbf{v}$ updated using a rank one update, analogous to standard Gaussian process classification [77]. A final issue worth noting is that we have a non standard prior on δ which is difficult to deal with. We approximate the prior on δ with another Gaussian factor. The moments required for computing the parameters of this Gaussian are estimated numerically. Since, δ is an unidimensional quantity, numerical moment computation is easy and efficient. Furthermore, these moments are required only once per EP sweep, where a sweep is defined as circling through all the super-pixels. Thus the added computational cost of numerical moment computation is negligible.

A.1.1 Computational Complexity

Observe that we only explicitly maintain a Gaussian posterior distribution on $\mathbf{v}$ which is a D dimensional quantity. Thus, the complexity of one EP sweep is $O(ND^2)$ as opposed to standard Gaussian process classification which has a complexity of $O(N^3)$ where N is the number of super-pixels. Observe that for any candidate partition, the prior for all layers can be evaluated in parallel. Thus, the cost of running T search iterations, each iteration running t sweeps of EP is $O(tTND^2)$.

A.2 Likelihood Evaluation

The likelihood computation involves evaluating the independent color and texture integrals

$$\int_{\Theta} p(\mathbf{x}|\boldsymbol{z},\Theta)p(\Theta|\rho)d\Theta = \int_{\theta^c} p(\mathbf{x^c}|\boldsymbol{z},\theta^c)p(\theta^c|\rho^c)d\theta^c \int_{\theta^t} p(\mathbf{x^t}|\boldsymbol{z},\theta^t)p(\theta^t|\rho^t)d\theta^t \tag{A.24}$$

which is a standard multinomial-Dirichlet integral. We provide the solution to the color integral here for the sake of completeness (*To simplify notation we denote θ^c, $\mathbf{x}^c$ by just θ and $\mathbf{x}$*).

For K segments and N super-pixels we have,

$$\int_{\theta} p(\mathbf{x}|\boldsymbol{z},\theta^c)p(\theta|\rho^c)d\theta = \prod_{k=1}^{K}\int_{\theta_k} p(\theta_k|\rho^c)\prod_{n=1}^{N} p(\boldsymbol{x}_n|z_n,\theta_k)^{\mathbb{I}(z_n=k)}d\theta_k \tag{A.25}$$

$$= \prod_{k=1}^{K}\int_{\theta_k} \Delta(\rho^c)\prod_{w=1}^{W_c}\theta_{kw}^{\rho^c_w-1}\prod_{n=1}^{N}\prod_{w=1}^{W_c}(\theta_{kw}^{x_{nw}})^{\mathbb{I}(z_n=k)}d\theta_k \tag{A.26}$$

$$= \prod_{k=1}^{K} \Delta(\rho^c) \int_{\theta_k} \prod_{w=1}^{W_c} \theta_{kw}^{\rho_w^c - 1} \prod_{w=1}^{W_c} (\theta_{kw})^{\sum_n x_{nw} \times \mathbb{I}(z_n = k)} d\theta_k \tag{A.27}$$

$$= \prod_{k=1}^{K} \Delta(\rho^c) \int_{\theta_k} \prod_{w=1}^{W_c} (\theta_{kw})^{x_w^k + \rho_w - 1} d\theta_k \tag{A.28}$$

$$= \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x^k)} \tag{A.29}$$

In the above derivation $\Delta(\rho^c) = \frac{\Gamma(\sum_w \rho_w^c)}{\prod_w \Gamma(\rho_w^c)}$ and x_w^k = number of times word w occurs with segment k. Putting it all together we have

$$\int_{\Theta} p(\mathbf{x}|\boldsymbol{z}, \Theta) p(\Theta|\rho) d\Theta = \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x_k^{(c)})} \frac{\Delta(\rho^t)}{\Delta(\rho^t + x_k^{(t)})} \tag{A.30}$$

A.3 Search Details

In this section we provide details of our search algorithm.

A.3.1 Shift move details

Notation note: *z_n is a categorical random variable assuming one of K values, where K is the number of components in the partition $\mathbf{z}$. t_n on the other hand is a binary random variable indicating whether super-pixel n is assigned to layer k or not. A is a N-by-D matrix, with rows $a_1^T \dots a_N^T$*

We are interested in optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ with respect to $\mathbf{z} = \{z_1, z_2 ... z_n\}$. In the shift move we assign each $z_n = \hat{k}$ such that $\hat{k} = \underset{k}{argmax}\ p(z_n = k \mid z_{-n}, \alpha, A, \Psi) p(\mathbf{x} \mid \mathbf{z}, \rho)$. Note that this implies we are optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ one z_n at a time.

1. for each super-pixel n

Algorithm 2: Search Pseudo-code

```
Get the initial partition $\mathbf{z}^0$ using $k$-means.
Set maxIter = 200, $i = 1$, bestMode = $\mathbf{z}^0$
while $i \leq$ maxIter do
   while $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ do
      Apply shift move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
      bestMode = $\mathbf{z}^i$
      $i = i + 1$
   end while
   if $i \leq$ maxIter then
      Select a move from the set { Merge, Swap, Split }
      Apply the selected move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
      if $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ then
         bestMode = $\mathbf{z}^i$
      end if
      $i = i + 1$
   end if
end while
return bestMode
```

(a) for each layer k

i. If super-pixel n is defined for layer k; Compute the approximate posterior cavity distribution on $\mathbf{v}$; $q(\mathbf{v}|\boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_{-n}, \Sigma_{-n})$ and the approximate posterior cavity distribution for the layer's threshold δ_k; $q(\delta_k|\boldsymbol{t}_{-n}) = \mathcal{N}(\delta_k|\mu_{-\delta_n}, \sigma^2_{-\delta_n})$

ii. If super-pixel n is not defined for layer k (ie it has already been assigned to a previous layer) the posterior distributions on $\mathbf{v}$ and δ_k are themselves the cavity distributions.

iii. Next, compute the parameters of the conditional distribution $q(u_n|\mathbf{v}, \boldsymbol{t}_{-n}) = q(u_n|\mu_*, \sigma^2{}_*)$, given by

$$\mu_* = a_n^T \boldsymbol{\mu}_{-n} \tag{A.31}$$

$$\sigma_*^2 = \Psi_n + a_n^T \Sigma_{-n} a_n \tag{A.32}$$

iv. Finally, compute $\pi_{nk} = p(t_n = 1|t_{-n})$ as follows

$$\begin{aligned}\pi_{nk} &= E_q[\mathbb{I}(u_n < \delta_k)] \\ &= \int \mathbb{I}(u_n < \delta_k)\mathcal{N}(u_n|\mu_*, \sigma_*^2)\mathcal{N}(\delta_k|\mu_{-\delta_n}, \sigma_{-\delta_n}^2)du_n d\delta_k \\ &= \Phi\left(\frac{\mu_{-\delta_n} - \mu_*}{\sqrt{\sigma_*^2 + \sigma_{-\delta_n}^2}}\right)\end{aligned}$$

v. The probability of super-pixel n getting assigned to layer k is given by

$$p(z_n = k \mid z_{-n}) = p(u_n < \delta_k \mid u_n > \delta_l) = \pi_{nk}\prod_{l=1}^{k-1}(1 - \pi_{nl}) \quad \text{(A.33)}$$

vi. Compute the posterior probability of the super-pixel assignment

$$p(\mathbf{z} \mid \mathbf{x}, \rho, \alpha) \propto p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta)p(\theta \mid \rho)d\theta \quad \text{(A.34)}$$

(b) Finally, assign n to layer $\hat{k}$ which maximizes posterior probability

$$\hat{k} = \underset{k}{\operatorname{argmax}}\, p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta)p(\theta \mid \rho)d\theta \quad \text{(A.35)}$$

(c) For all layers affected by the shift of super-pixel n, update the corresponding posterior distribution on $\mathbf{v}$ by a EP projection for the relevant super-pixel. Care is taken such that when a previously invalid super-pixel gets shifted into a layer, the old posterior is treated as the new cavity distribution. Likewise when a super-pixel is shifted out of a layer, the old cavity distribution is treated as the new posterior.

Appendix B

Marginal Likelihoods for Chapter 4

Let $Y = [\mathbf{y}_1, ...\mathbf{y}_N] \in R^{3\times N}$ denote the coordinates of mesh faces assigned to the same part in a given pose. Let $X = [\mathbf{x}_1, ...\mathbf{x}_N] \in R^{4\times N}$ represent the corresponding reference (homogeneous) coordinates. The distribution of $Y|X$ (for a given part and pose) is then given by

$$Y|X \sim \mathcal{MN}(\mathcal{A}X, \Sigma, \mathbf{I}) \tag{B.1}$$

From [111] - F.10 we have

$$p(Y|X,\Sigma) = \int p(Y,\mathcal{A}|X,\Sigma)d\mathcal{A} = \frac{|K|^{3/2}}{|2\pi\Sigma|^{N/2}|S_{xx}|^{3/2}}exp\{-\frac{1}{2}tr(\Sigma^{-1}S_{y|x})\} \tag{B.2}$$

and

$$S_{xx} = XX^T + K \tag{B.3}$$

$$S_{yx} = YX^T + MK \tag{B.4}$$

$$S_{y|x} = YY^T + MKM^T - S_{yx}(S_{xx})^{-1}S_{yx}^T \tag{B.5}$$

Finally, the marginal likelihood is given by

$$\begin{aligned}\mathcal{N}(p(Y|X) \quad &= \textstyle\int p(Y|X,\Sigma)p(\Sigma|n_0,S_0)d\Sigma \quad &\text{(B.6)}\\ &= \textstyle\int \frac{|K|^{3/2}}{|2\pi\Sigma|^{3N/2}|S_{xx}|^{3/2}} exp\{-\frac{1}{2}tr(\Sigma^{-1}S_{y|x})\} \quad &\text{(B.7)}\\ &\textstyle\frac{|S_0|^{n_0/2}|\Sigma|^{-(4+n_0)/2}}{2^{3n_0/2}\Gamma_3(n_0/2)} exp\{-\frac{1}{2}tr(\Sigma^{-1}S_0)\}d\Sigma \quad &\text{(B.8)}\end{aligned}$$

$$p(Y|X) = \int \frac{|K|^{3/2}|S_0|^{n_0/2}|\Sigma|^{-(4+n_0)/2}}{|2\pi\Sigma|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)} exp\{-\frac{1}{2}tr(\Sigma^{-1}(S_{y|x}+S_0))\}d\Sigma \tag{B.9}$$

$$p(Y|X) = \frac{|K|^{3/2}|S_0|^{n_0/2}}{|2\pi|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)} \int |\Sigma|^{-(3+N+n_0+1)/2} \\ exp\{-\frac{1}{2}tr(\Sigma^{-1}(S_{y|x}+S_0))\} \tag{B.10}$$

$$p(Y|X) = \frac{|K|^{3/2}|S_0|^{n_0/2}2^{(N+n_0)3/2}\Gamma_3((N+n_0)/2)}{|2\pi|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)|S_0+S_{y|x}|^{(N+n_0)/2}} \int IW(N+n_0, \\ S_{y|x}+S_0)d\Sigma \tag{B.11}$$

The marginal likelihood for one part in one pose is then given by

$$p(Y|X,K,n_0,S_0) = \frac{|K|^{\frac{3}{2}}|S_0|^{\frac{n_0}{2}}\Gamma_3\left(\frac{N+n_0}{2}\right)}{\pi^{\frac{3N}{2}}|S_{xx}|^{\frac{3}{2}}|S_0+S_{y|x}|^{\frac{(N+n_0)}{2}}\Gamma_3(\frac{n_0}{2})} \tag{B.12}$$

Appendix C

MCMC Details from Chapter 5

C.1 Inference Details

Algorithm 3: Iterative sampling of customer and table links.

```
for $i \in 1 \dots N$ do
    $\mathbf{c}^*, \mathbf{k}^* \longleftarrow$ CustLinkProposal($i, \mathbf{x}, \mathbf{k}, \mathbf{c}, \alpha, D, \alpha_0, A^0(\mathbf{c})$)
    Compute acceptance ratio $\rho$ ;                                   /*See supplement*/
    With probability $\propto \min(1, \rho)$, accept $\mathbf{c}, \mathbf{k} \longleftarrow \mathbf{c}^*, \mathbf{k}^*$
for $t \in T(\mathbf{c})$ do
    $k_t \sim p(k_t \mid \mathbf{k}_{-t}, \mathbf{c}, \mathbf{x}, \alpha_0, A^0(\mathbf{c}))$ ;      /*Gibbs update $k_t$*/
```

Algorithm 4: CustLinkProposal

```
input  : $i, \mathbf{x}, \mathbf{k}, \mathbf{c}, \alpha, D, \alpha_0, A^0(\mathbf{c})$
output: $\mathbf{k}^*, \mathbf{c}^*, q_{\cdot}(\mathbf{c}^*, \mathbf{k}^* \mid \mathbf{k}, \mathbf{c})$
$i' \longleftarrow c_i$
Set $c_i = i$ and update $\mathbf{c}^o = \{c_1, \ldots, c_{i-1}, c_i = i, \ldots, c_N\}$;
$L_{t_i} = \{t_\ell \mid k_{t_\ell} = t_i \;\&\; t_\ell \neq t_i\}$ ;  /*Set of all tables pointing to $t_i$, except
self loops.*/
if A new table is created by setting $c_i = i$ then
    Set split = true;                       /*Record the occurrence of a split.*/
    $\mathbf{k} \longleftarrow$ ReassignLinks ($L_{t_i}$)
    $k^*_{t_{i'}} \longleftarrow k_{t_{i,i'}}$ ;   /*A split table retains the current table's link.*/
Sample $c_i^* \sim q(c_i^*)$
if $c_i^*$ causes two existing tables to merge then
    Set $t_{i,i^*} = t_i \cup t_{i^*}$
    $L_{t_{i,i^*}} = L_{t_i} \cup L_{t_{i^*}}$ and Update $\mathbf{k}$ to reflect the merge
    if split then                                           /* Split+Merge */
        $k^*_{t_{i,i^*}} \longleftarrow k_{t_{i^*}}$
        $q_{sm}(\mathbf{c}^*, \mathbf{k}^* \mid \mathbf{c}, \mathbf{k}, X) = (0.5)^{|L_{t_i}|} q(c_i^*)$
    else                                                /* No split + Merge */
        Delete $k_{t_i}$
        $k^*_{t_{i,i^*}} \longleftarrow k_{t_{i^*}}$
        $q_m(\mathbf{c}^*, \mathbf{k}^* \mid \mathbf{c}, \mathbf{k}, X) = q(c_i^*)$
else
    if split then                                        /* Split+No Merge */
        Sample $k^*_{t_i} \sim p(k^*_{t_i} \mid \alpha_0, A^0(\mathbf{c})(\mathbf{c}^*), \mathbf{x}, \mathbf{k}_{-t_i})$ ;
        $q_s(\mathbf{c}^*, \mathbf{k}^* \mid \mathbf{c}, \mathbf{k}, X) = (0.5)^{|L_{t_i}|} q(c_i^*) p(k^*_{t_i} \mid A^0(\mathbf{c})(\mathbf{c}^*), \mathbf{x}, \mathbf{k}^*_{-t_i})$
    else                                              /* No Split+No Merge */
        /*No change to partition - Do Nothing.                           */
        $q_{nc}(\mathbf{c}^*, \mathbf{k}^* \mid \mathbf{c}, \mathbf{k}, X) = q(c_i^*)$
```

Algorithm 5: ReassignLinks

```
input  : $L_{t_i}$
output: $\mathbf{k}$
/*Reassign links pointing to a split table.  Links are assigned to
  one of the two split tables.                                      */
for $\ell \in L_{t_i}$ do
    $b_\ell \sim$ Ber(0.5)
    if $b_\ell = 1$ then
        $L_{t_i} = L_{t_i} / t_\ell$
        $L_{t_{i'}} = L_{t_{i'}} \cup t_\ell$
        $k_{t_j} = t_{i'}$;
```

Bibliography

[1] E Wachsmuth, MW Oram, and DI Perrett. Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque. *Cerebral Cortex*, 4(5):509–522, 1994.

[2] Nikos K Logothetis and David L Sheinberg. Visual object recognition. *Annual review of neuroscience*, 19(1):577–621, 1996.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012.

[4] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2008.4587503.

[5] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. In *ACM SIGGRAPH Asia 2009*

Papers, SIGGRAPH Asia '09, pages 124:1–124:10, New York, NY, USA, 2009. ACM.

[8] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *PAMI*, 24(8):1026–1038, August 2002.

[9] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer Graphics Forum*, volume 21, pages 219–228. Wiley Online Library, 2002.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, July 2001.

[11] Erik B. Sudderth and Michael I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1585–1592. 2008.

[12] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, September 2007.

[13] Chunhui Gu, Joseph J Lim, Pablo Arbelaez, and Jitendra Malik. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037. IEEE, 2009.

[14] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *Computer Vision, 1998. Sixth International Conference on*, pages 1154–1160. IEEE, 1998.

[15] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439. IEEE, June 2010.

[16] Richard Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

[17] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8), 2000.

[18] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. ISSN 0920-5691.

[19] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, May 2002.

[20] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[21] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, 1993.

[22] Erik B. Sudderth and Michael I. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *NIPS*, pages 1585–1592, 2008.

[23] X. Ren and J. Malik. A probabilistic multi-scale model for contour completion based on image statistics. In *ECCV*, volume 1, pages 312–327, 2002.

[24] Lawrence G. Roberts. *Machine Perception of Three-Dimensional Solids.* Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963. ISBN 0-8240-4427-4.

[25] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

[26] D. R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004.

[27] Pierre Parent and Steven W Zucker. Trace inference, curvature consistency, and curve detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(8):823–839, 1989.

[28] Lance R Williams and David W Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9 (4):837–858, 1997.

[29] James H Elder and Steven W Zucker. Computing contour closure. In *Computer VisionECCV'96*, pages 399–412. Springer, 1996.

[30] Xiaofeng Ren, Charless C Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1214–1221. IEEE, 2005.

[31] Pedro Felzenszwalb and David McAllester. A min-cover approach for finding salient curves. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 185–185. IEEE, 2006.

[32] S. Beucher and Centre De Morphologie Mathmatique. The watershed transformation applied to image segmentation. In *Scanning Microscopy International*, pages 299–314, 1991.

[33] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[34] Marco Andreetto, Lihi Zelnik-Manor, and Pietro Perona. Non-parametric probabilistic image segmentation. In *ICCV*, 2007.

[35] Giorgos Sfikas, Christophoros Nikou, and Nikolaos Galatsanos. Edge preserving spatially varying mixtures for image segmentation. *CVPR*, 0: 1–7, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2008.4587416.

[36] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6(6):721–741, 1984.

[37] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In *Computer Vision-ECCV 2004*, pages 428–441. Springer, 2004.

[38] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[39] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

[40] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[41] Brian L Price, Bryan Morse, and Scott Cohen. Geodesic graph cut for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3161–3168. IEEE, 2010.

[42] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 277–284. IEEE, 2009.

[43] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, November 2001.

[44] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-007-0090-8.

[45] W. M. Rand. Objective criteria for the evaluation of clustering methods. *JASA*, 66(336):846–850, 1971.

[46] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. PAMI*, 29(6):929–944, 2007.

[47] M. Meila. Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[48] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, pages 1605–1614, 2006.

[49] RD Morris, X Descombes, and J Zerubia. The ising/potts model is not well suited to segmentation tasks. In *Digital Signal Processing Workshop Proceedings, 1996., IEEE*, pages 263–266. IEEE, 1996.

[50] Peter Orbanz and Joachim M Buhmann. Nonparametric bayesian image segmentation. *International Journal of Computer Vision*, 77(1-3):25–45, 2008.

[51] Lu Ren, Lan Du, Lawrence Carin, and David Dunson. Logistic stick-breaking process. *The Journal of Machine Learning Research*, 12:203–239, 2011.

[52] S. Ghosh, A. B. Ungureanu, E. B. Sudderth, and D. Blei. Spatial distance dependent Chinese restaurant processes for image segmentation. In *NIPS*, pages 1476–1484, 2011.

[53] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. In *CVPR*, 2010.

[54] Sylvain Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *Computer Vision–ECCV 2008*, pages 460–473. Springer, 2008.

[55] Hayit Greenspan, Jacob Goldberger, and Arnaldo Mayer. A probabilistic framework for spatio-temporal video representation & indexing. In *Computer VisionECCV 2002*, pages 461–475. Springer, 2002.

[56] David Tsai, Matthew Flagg, and James M.Rehg. Motion coherent tracking with multi-label mrf optimization. *BMVC*, 2010.

[57] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[58] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(6):1187–1200, 2014.

[59] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Computer Vision–ECCV 2010*, pages 282–295. Springer, 2010.

[60] José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.

[61] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[62] B. Frey N. Jojic and A.Kannan. Learning appearance and transparency manifolds of occluded objects in layers. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[63] Anitha Kannan, Nebojsa Jojic, and B Frey. Generative model for layers of appearance and deformation. *AIStats 05*, 2005.

[64] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. In *Proceedings of the International Conference on Computer Vision*, 2005.

[65] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775, 2012.

[66] Chenliang Xu, Caiming Xiong, and Jason J Corso. Streaming hierarchical video segmentation. In *Computer Vision–ECCV 2012*, pages 626–639. Springer, 2012.

[67] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *International Conference on Computer Vision (ICCV)*, December 2013.

[68] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation — A comparative study. In *SMI*, 2006.

[69] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3):73:1–73:12, 2009.

[70] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics*, 29(4): 102:1–102:12, July 2010.

[71] Ariel Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.

[72] D. Anguelov, D. Koller, H. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3d range data. In *UAI*, pages 18–26, 2004.

[73] Tong-Yee Lee, Yu-Shuen Wang, and Tai-Guang Chen. Segmenting a deforming mesh into near-rigid components. *The Visual Computer*, 22(9):729–739, September 2006. ISSN 0178-2789.

[74] Guy Rosman, Michael M. Bronstein, Alexander M. Bronstein, Alon Wolf, and Ron Kimmel. Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes. In *SSVM'11*, pages 725–736, 2012. ISBN 978-3-642-24784-2.

[75] Stefanie Wuhrer and Alan Brunton. Segmenting animated objects into near-rigid components. *The Visual Computer*, 26:147–155, 2010. ISSN 0178-2789.

[76] J. Franco and E. Boyer. Learning temporally consistent rigidities. In *IEEE CVPR*, pages 1241–1248, 2011.

[77] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. ISBN 026218253X.

[78] C. E. Antoniak. Smooth image segmentation by nonparametric bayesian inference. *ECCV*, 1(6):1152 – 1174, 1974.

[79] J. Pitman. *Combinatorial Stochastic Processes*. Lecture Notes for St. Flour Summer School. Springer-Verlag, New York, NY, 2002.

[80] Charles Blundell, Yee Whye Teh, and Katherine A Heller. Bayesian rose trees. *arXiv preprint arXiv:1203.3468*, 2012.

[81] Thomas L Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12: 1185–1224, 2011.

[82] T. Broderick, L. Mackey, J. Paisley, and M.I. Jordan. Combinatorial clustering and the beta negative binomial process. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):290–306, Feb 2015.

[83] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2410–2417. IEEE Computer Society, 2006.

[84] Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

[85] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2010.

[86] Michael I Jordan. Bayesian nonparametric learning: Expressive priors for intelligent systems. *Heuristics, Probability and Causality: A Tribute to Judea Pearl*, 11:167–185, 2010.

[87] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York.

[88] Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9: 2035–2078, 2008.

[89] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *UAI*, pages 362–369, 2001.

[90] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[91] David Blackwell. Discreteness of ferguson selections. *The Annals of Statistics*, 1(2):356–358, 1973.

[92] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639 – 650, 1994.

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

[94] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *JCGS*, 9(2):249–265, 2000.

[95] Richard Nock and Frank Nielsen. Statistical region merging. *PAMI*, 26: 1452–1458, November 2004. ISSN 0162-8828.

[96] Alex Shyr, Trevor Darrell, Michael I. Jordan, and Raquel Urtasun. Supervised hierarchical Pitman-Yor process for natural scene segmentation. In *CVPR*, pages 2281–2288, 2011.

[97] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, volume 1, pages 10–17, 2003.

[98] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26:530–549, May 2004. ISSN 0162-8828.

[99] J. Pitman and M. Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.

[100] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Tran. IP*, 3(5):625–638, September 1994.

[101] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *IJCV*, 72(2):195–215, 2007.

[102] M. Welling and K. Kurihara. Bayesian K-means as a “Maximization-Expectation” algorithm. In *SDM*, 2006.

[103] Z. Tu and S.C. Zhu. Image segmentation by data-driven Markov Chain Monte Carlo. *PAMI*, 24:657–673, 2002. ISSN 0162-8828.

[104] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic Photo Pop-up. In *ACM SIGGRAPH*, SIGGRAPH '05, pages 577–584, 2005.

[105] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[106] Rdiger Borsdorf, Nicholas J. Higham, and Marcos Raydan. Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Analysis App.*, 31(5):2603–2622, 2010.

[107] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[108] D. M. Blei and P. I. Frazier. Distance dependent Chinese restaurant processes. *J. Mach. Learn. Res.*, 12:2461–2488, November 2011.

[109] D. M. Blei and P. I. Frazier. Distant dependent chinese restaurant process. *arXiv:0910.1022v1*, 2009.

[110] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *ECCV*, pages 242–255, 2012.

[111] E. B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[112] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions.* Chapman & Hall/CRC, October 2000. ISBN 1584880465.

[113] Richard D. De Veaux. Mixtures of linear regressions. *Computational Statistics and Data Analysis*, 8(3):227 – 245, 1989. ISSN 0167-9473.

[114] Lauren Hannah, David M Blei, and Warren B Powell. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research*, 12:1923–1953, 2011.

[115] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum (Proc. Eurographics 2009)*, volume 2, pages 337–346, March 2009.

[116] R. N. Shepard. Multidimensional scaling, tree-fitting, and clustering. *Science*, 210:390–398, October 1980.

[117] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE PAMI*, 33 (3):568–586, 2011.

[118] Rong Liu and Hao Zhang. Segmentation of 3D meshes through spectral clustering. In *Pacific Conference on Computer Graphics and Applications*, pages 298–305, 2004.

[119] Edilson de Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum*, 27(2):389–397, 2008.

[120] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. Calculus of nonrigid surfaces for geometry and texture manipulation. *IEEE Tran. on Viz. and Computer Graphics*, 13:902–913, 2007. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.1041.

[121] Oren Freifeld and Michael J. Black. Lie bodies: A manifold representation of 3D human shape. In *European Conf. on Computer Vision (ECCV)*, Part I, LNCS 7572, pages 1–14. Springer-Verlag, October 2012.

[122] J. Pitman. Combinatorial stochastic processes. Technical Report 621, U.C. Berkeley Department of Statistics, August 2002.

[123] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of American Statistical Association*, 25(2):1566–1581, 2006.

[124] D. Kim and A. Oh. Accounting for data dependencies within a hierarchical Dirichlet process mixture model. In *CIKM*, pages 873–878, 2011.

[125] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[126] S. Ghosh, M. Raptis, L. Sigal, and E. B. Sudderth. Nonparametric clustering with distance dependent hierarchies. In *UAI*, 2014.

[127] E.B. Fox, M.C. Hughes, E.B. Sudderth, and M.I. Jordan. Joint modeling of multiple related time series via the beta process with application to motion capture segmentation. *Annals of Applied Statistics*, 8(3):1281–1313, 2014.

[128] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, GI '04, pages 185–194. Canadian Human-Computer Communications Society, 2004. ISBN 1-56881-227-2.

[129] M. Riedl and C. Biemann. How text segmentation algorithms gain from topic models. In *HLT-NAACL*, pages 553–557, 2012.

[130] H. Chen, S. R. K. Branavan, R. Barzilay, and D. R. Karger. Content modeling using latent permutations. *J. Artif. Intell. Res. (JAIR)*, 36:129–163, 2009.

[131] Alison Wray. *Formulaic language and the lexicon*, volume 5. 2002.

[132] S. Ghosh, E. B. Sudderth, M. Loper, and M. J. Black. From deformations to parts: Motion-based segmentation of 3D objects. In *NIPS*, pages 2006–2014, 2012.

[133] W. Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[134] Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. Pos induction with distributional and morphological information using a distance-dependent chinese restaurant process. In *Proceedings of the 52nd Annual Meeting of the Association of Computational Linguistics, Volume 2: Short Papers*, 2014.

[135] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2: 54–61, 2003.

[136] X. Ren and J. Malik. Learning a classification model for segmentation. *International Conference on Computer Vision (ICCV)*, 2003.

[137] Francis R Bach and Michael I Jordan. Learning spectral clustering, with application to speech separation. *The Journal of Machine Learning Research*, 7:1963–2001, 2006.

[138] Kevin Briggman, Winfried Denk, Sebastian Seung, Moritz N Helmstaedter, and Srinivas C Turaga. Maximin affinity learning of image segmentation. In *Advances in Neural Information Processing Systems*, pages 1865–1873, 2009.

[139] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[140] X. Ren and J. Malik. Learning a classification model for segmentation. *ICCV*, 2003.

[141] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145 – 175, 2001.

[142] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database web-based tool for image annotation. *IJCV*, 77:157–173, 2008.

[143] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[144] G. Mori. Guiding model search using segmentation. *ICCV*, 2005.

[145] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *CVPR*, 2:54–61, 2003.

[146] D. R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004.

[147] J. Chang, D. Wei, and J. W. Fisher III. A Video Representation Using Temporal Superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[148] Fabio Galasso, Roberto Cipolla, and Bernt Schiele. Video segmentation with superpixels. In *Asian Conference on Computer Vision*, 2012.

[149] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*, pages 589–600. Springer, 2006.

[150] Anoop Cherian, Vassilios Morellas, Nikolaos Papanikolopoulos, and Saad J Bedros. Dirichlet process mixture models on symmetric positive definite matrices for appearance clustering in video surveillance applications. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3417–3424, 2011.

[151] Sergey Bartunov and Dmitry Vetrov. Variational inference for sequential distance dependent chinese restaurant process. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1404–1412, 2014.