UNIVERSITY OF CALIFORNIA,
IRVINE

Efficient Variational Inference for Hierarchical Models of Images, Text, and Networks

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Geng Ji

Dissertation Committee:
Professor Erik Sudderth, Chair
Professor Alexander Ihler
Professor Stephan Mandt

2019

DEDICATION

To my great grandma and grandma who passed away during my graduate study in the U.S.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Page

ACKNOWLEDGMENTS

First of all, I would like to thank my great advisor, Professor Erik Sudderth, who patiently led me through the entire PhD journey. Looking back, the most impressive thing I can remember is the big smile on his face every time I talk to him. The way he guides me in research is through the trigger of my interests in exploring novel things. When my braveness is used up, he recharges me by providing helpful suggestions based on his wide knowledge and deep understanding of the whole area. He is also a very caring person in life, sweetly moving to the beautiful UC Irvine after hearing my occasional complaints about the cold winters in New England and the hard access to high-quality Asian food. I bet he'll agree with this reason for moving, just like he always claims that the only reason for his coming to the office super early every day is that his young kids regularly wake him up at six.

I would like to express my sincere gratitude to my thesis committee members, Professor Alex Ihler and Stephan Mandt. Their help and support on my graduate study have also gone beyond my 2.5 years of PhD life at UC Irvine. Alex was the instructor of the first machine learning course I ever took. The knowledge I learned from his classes and the weekly AI/ML seminars he organized during my MS study eventually led me to a PhD program of this exciting area. Back on the east coast I once thought about inviting him as an outside committee member. But life is just like a box of chocolate – now he directly is, and one can never imagine how much help I've got from him after coming back. Stephan was my mentor at Disney Research, where I did the first summer internship of my life. The hands-on experience I got there in probabilistic programming languages unexpectedly leads to one of the research topics of this thesis. From the rooftop bar near Carnegie Mellon University, to the wax museum down in Sydney, and then the happy reunion in Southern California, I hope my friendship with him could proceed in more and more unforgettable places.

I also feel very grateful to all the labmates of our Learning, Inference and Vision group. Lab pioneers Soumya Ghosh, Michael Hughes, Daeil Kim and Jason Pacheco set great examples to me of what self-motivated PhD students should be like. Although the length of my overlap with them is not very long, the spiritual power I experienced from these role models always make me feel stronger during the years later when in front of research difficulties. Zhile Ren and Gabriel Hope are the ones that I spend most of the years with in the group. I'm quite glad to have them to discuss with. They inspire me all the time, from coast to coast. I also enjoy interacting with junior members Debora Sujono, Harry Bendekgey and Xiaoyin Chen, from whom I see a bright future of the LIV lab that I will always belong to.

I would also like to thank all the other residents of the DBH building at UC Irvine, and the CIT building at Brown, who spent days and nights with me together during my graduate life. I'm also super grateful to the wonderful colleagues I met at Disney Research and Facebook AI during the summer internships.

Finally, I want to thank my parents, parents-in-law and my wife from the very bottom of my heart for their unconditional support for me pursuing the PhD degree on both coasts throughout the years. To them I am forever indebted.

CURRICULUM VITAE

Geng Ji

EDUCATION

Doctor of Philosophy in Computer Science **2019**
University of California, Irvine *Irvine, California*

Master of Science in Computer Science **2017**
Brown University *Providence, Rhode Island*

Master of Science in Computer Science **2014**
University of California, Irvine *Irvine, California*

Bachelor of Engineering in Engineering Physics **2012**
Tsinghua University *Beijing, China*

INDUSTRY EXPERIENCE

Research Intern **06/2018 – 09/2018**
Facebook AI *Menlo Park, California*

R&D Lab Intern **05/2017 – 08/2017**
Disney Research *Pittsburgh, Pennsylvania*

TEACHING

UCI CS 274B Learning in Graphical Models Spring 2018
UCI CS 177 Applications of Probability in Computer Science Fall 2017
Brown CSCI 1420 Introduction to Machine Learning Fall 2015

ACADEMIC REVIEWING

AAAI Conference on Artificial Intelligence (AAAI) 2020
Conference on Neural Information Processing Systems (NeurIPS) 2019
International Conference on Machine Learning (ICML) 2019
Bayesian Deep Learning Workshop at NeurIPS 2019
Practical Bayesian Nonparametrics Workshop at NeurIPS 2018
Symposium on Advances in Approximate Bayesian Inference 2018

PUBLICATIONS

Geng Ji, Dehua Cheng, Huazhong Ning, Changhe Yuan, Hanning Zhou, Liang Xiong, and Erik B. Sudderth. Variational Training for Large-Scale Noisy-OR Bayesian Networks. *Uncertainty in Artificial Intelligence (UAI)*, 2019.

Geng Ji, Michael C. Hughes, and Erik B. Sudderth. From Patches to Images: A Nonparametric Generative Model. *International Conference on Machine Learning (ICML)*, 2017.

Geng Ji, Robert Bamler, Erik B. Sudderth, and Stephan Mandt. Bayesian Paragraph Vectors. *Advances in Approximate Bayesian Inference Workshop at NeurIPS*, 2017.

Geng Ji, Michael C. Hughes, and Erik B. Sudderth. From Patches to Images via Hierarchical Dirichlet Process. *Practical Bayesian Nonparametrics Workshop at NeurIPS*, 2016.

ABSTRACT OF THE DISSERTATION

Efficient Variational Inference for Hierarchical Models of Images, Text, and Networks

By

Geng Ji

Doctor of Philosophy in Computer Science

University of California, Irvine, 2019

Professor Erik Sudderth, Chair

Variational inference provides a general optimization framework to approximate the posterior distributions of latent variables in probabilistic models. Although effective in simple scenarios, variational inference may be inaccurate or infeasible when the data is high-dimensional, the model structure is complicated, or variable relationships are non-conjugate. We propose solutions to these problems through the smart design and leverage of model structures, the rigorous derivation of variational bounds, and the creation of flexible algorithms for various models with rich, non-conjugate dependencies.

Concretely, we first design an interpretable generative model for natural images, in which the hundreds of thousands of pixels per image are split into small patches represented by Gaussian mixture models. Through structured variational inference, the evidence lower bound of this model automatically recovers the popular expected patch log-likelihood method for image processing. A nonparametric extension using hierarchical Dirichlet processes further enables self-similarities to be captured and image-specific clusters created during inference, boosting image denoising and inpainting accuracy.

Then we move on to text data, and design hierarchical topic graphs that generalize the bipar-

tite noisy-OR models previously used for medical diagnosis. We derive auxiliary bounds to overcome the non-conjugacy of noisy-OR conditionals, and use stochastic variational inference to efficiently train on datasets with hundreds of thousands of documents. We dramatically increase the algorithm speed through a constrained family of variational bounds, so that only the ancestors of the sparse observed tokens of each document need to be considered.

Finally, we propose a general-purpose Monte Carlo variational inference strategy that is directly applicable to any model with discrete variables. Compared to REINFORCE-style stochastic gradient updates, our coordinate-ascent updates have lower variance and converge much faster. Compared to auxiliary-variable bounds crafted for each individual model, our algorithm is simpler to derive and may be easily integrated into probabilistic programming languages for broader use. By avoiding auxiliary variables, we also tighten likelihood bounds and increase robustness to local optima. Extensive experiments on real-world models of images, text, and networks illustrate these appealing advantages.

Chapter 1

Introduction

Probabilistic graphical models are known as an elegant tool to describe the dependency relations between variables in high-dimensional data [Wainwright and Jordan, 2008]. Previous work has shown that inference queries on large-scale, complicated models could be effectively approximated through variational methods, whereas exact or sampling-based inference is often computationally infeasible [Jordan et al., 1999, Jaakkola and Jordan, 1999, Beal and Ghahramani, 2006, Gopalan and Blei, 2013, Gan et al., 2015, Gopalan et al., 2016]. In this chapter, we first do a quick review of the basics of probabilistic models and the fundamental ideas behind variational inference, with a focus on topics the thesis relevant to. Then we discuss the challenges remaining in this area, and introduce our contributions that empower efficient variational algorithms for a wide range of interpretable hierarchical models of image, text and network data.

1.1 Background

1.1.1 Directed Probabilistic Models

Directed probabilistic models are also known as Bayesian or belief networks. Take $p(z, x) = p(z)p(x \mid z)$ as a simple example. Latent (or hidden) variables z usually correspond to high-

level objects with human-interpretable meanings, such as the cluster assignments in mixture models, or the unobserved chain of state transitions for each sequence in hidden Markov models (HMMs). $p(z)$ describes the *prior* knowledge about the distribution of values that z may take, before any data evidence is given.

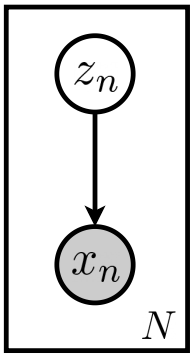


Figure 1.1: A toy example of directed probabilistic models. Shaded nodes are observed. Plates indicate replicated variables.

We assume the observed data x is generated by samples drawn from the *likelihood* function $p(x \mid z)$. This sequential process can be represented by the directed graphical model shown in Figure 1.1, in which z serves as the parents of x. The square box surrounding the variables indicates another assumption made, which is data points x_n are independent to each other given z_n, for $n = 1, \cdots, N$. Note that this kind of *conditional independence* may further exist at finer scales, where each of the variables in x_n only depends on a subset of variables in z_n. For example, given the parent states of an HMM sequence, the observations across time steps become all independent to each other. Structural information like this simplifies the dependencies between variables, and is also one of the keys properties we will leverage in this dissertation for building efficient variational algorithms.

1.1.2 Mean-field Variational Inference

Given the data observation x, we're interested in the *posterior* probability $p(z \mid x)$, which describes how the latent patterns of the model explain the data observations. In probability,

tasks like this are generally called *inference*. According to Bayes rule, we have

$$p(z \mid x) = \frac{p(z, x)}{p(x)} = \frac{p(z, x)}{\int p(z, x) \mathrm{d}z}. \tag{1.1}$$

While the numerator $p(z, x)$ is easy to evaluate given the generative process, the data marginal $p(x)$ in the denominator, also known as the *evidence*, is often intractable to compute. Mean-field *variational inference* (VI) thus seeks to find a variational distribution $q(z)$ that minimizes the Kullback–Leibler (KL) divergence to this true posterior

$$D_{\mathrm{KL}}\Big(q(z) \parallel p(z \mid x)\Big) = \mathbb{E}_{q(z)}\bigg[\log \frac{q(z)}{p(z \mid x)}\bigg] = \int \Big[\log q(z) - \log p(z \mid x)\Big] q(z) \mathrm{d}z. \tag{1.2}$$

Because the KL divergence is always nonnegative, minimizing Equation (1.2) is equivalent to maximizing the *evidence lower bound* (ELBO, Jordan et al. [1999])

$$\begin{aligned} \mathcal{L}\big(q(z)\big) &\triangleq \mathbb{E}_{q(z)}\Big[\log p(z, x) - \log q(z)\Big] \\ &= \log p(x) - D_{\mathrm{KL}}\Big(q(z) \parallel p(z \mid x)\Big) \leq \log p(x), \end{aligned} \tag{1.3}$$

in which equality is achieved if and only if $q(z)$ is identical to the true posterior $p(z \mid x)$ (exact inference). To make variational inference useful in practice, one typically restricts $q(z)$ to a family of simpler distributions $\mathcal{Q}$, and look for the best approximation within it. The simplest and most common choice is the *naïve* mean field, where $q(z)$ is fully factorized across each hidden variable. In this thesis, we also consider *structured* mean field, which tightens the evidence lower bound by enabling variational dependencies.

1.2 Graphical Models for High-dimensional Data

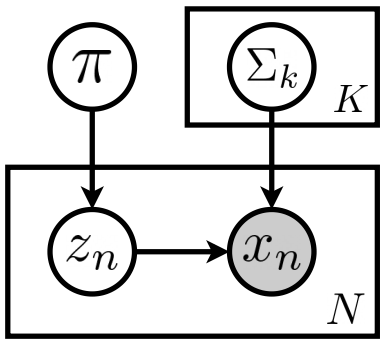
In this section, we show a couple of graphical models that capture dependencies within different kinds of high-dimensional data. These models will be further extended and explored using the variational inference algorithms developed in the next few chapters.

1.2.1 Gaussian Mixture Models for Natural Image Patches

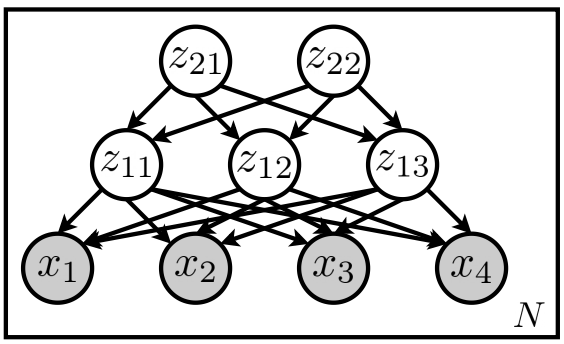
Gaussian mixture models (GMMs) have been found to perform well in capturing the density of natural image patches [Zoran and Weiss, 2012], such as the isotropic textures in sky and trees, and the straight lines and sharp corners observed in buildings and indoor scenes. As shown in Figure 1.2(a), a zero-mean GMM could be represented as $p(x_n) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x \mid 0, \Sigma_k)$, in which π_k is the probability of choosing cluster k, and Σ_k is the corresponding covariance matrix. For each zero-centered patch x_n, variable $z_n \in [1, K]$ indicates the cluster assignment.

When we observe a corrupted version of the image patch y_n and want to restore it, this GMM prior $p(x_n)$ could help us infer the posterior distribution $p(x_n \mid y_n)$ as the output. According to Bayes rule, $p(x_n \mid y_n) \propto p(x_n)p(y_n \mid x_n)$. The likelihood term $p(y_n \mid x_n)$ corresponds to the corruption process, such as adding white noise for denoising or convolving with a motion kernel in deblurring.

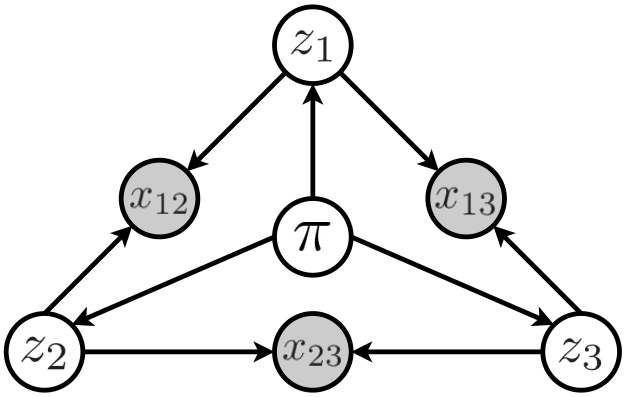
Compared to restoring small image patches, reconstructing whole natural images from corrupted observations is much more challenging due to their high dimensionality. We will discuss how to solve this problem in this dissertation, using the GMM of patches as a building block.



(a) Zero-mean Gaussian mixture model.



(b) Deep Bayesian network.



(c) Stochastic block model.

Figure 1.2: Graphical models for high-dimensional data of images, text and networks.

1.2.2 Deep Bayesian Networks for Binarized Documents and Images

Another type of graphical model we will explore is deep Bayesian networks. An example is shown in Figure 1.2(b). Like in regular fully-connected deep neural networks (DNNs), the connection between each layer is a linear combination of the previous layer's values, followed by a non-linear transform. But the difference with DNNs is that each node, both the latent ones z and the observed ones x, are random binary variables. Their activation probabilities are determined by the nonlinear transforms, of which a typical choice is the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$. We will also explore other types of nonlinearity in this thesis.

We will apply these type of models to binary data like the appearance of words in text documents, or the black and white pixels in binarized images. The hidden layers at the top are designed so that learned models capture high-level information, such as the presence of

different themes in news postings, or the styles and strokes in hand-written digits.

1.2.3 Relational Models for Communities and Networks

The final type of models we would explore in this dissertation focuses on the links in social networks. In cases where the connections are fully observed, each pair of entities in the network either has a link between them or the other way around. The kinds of relations we will encounter varies from the academic collaborations between people, to the co-participations of international conferences among countries. The goal of inference is to discover useful attributes for these entities from their interactions between one another, such as the communities to which they belong.

Figure 1.2(c) illustrates a simple relational model, where each entity i is a member of a single community indicated by z_i. Similar to the cluster allocations in mixture models, π controls the probability of community assignments. Variables x denote the presence or absence of connections, the number of which scales quadratically with the number of entities. For simplicity, only three entities are plotted in the toy network, but in real datasets the size could be huge. The probability of forming a link between i and j is decided by the global parameter $W_{z_i z_j}$, which depends on the communities they each come from and can be learned from data. In this dissertation, we will explore the inference problem in latent feature relational models by Miller et al. [2009], which enable more flexible attributes for each entity, and build up connections in more complicated ways.

1.3 Contributions Outline

While variational inference provides a general optimization framework to approximate the posterior distributions of latent variables, its performance may vary from model to model. In simple scenarios, such as in conjugate models where the expectations in Equation (1.3)

could all be easily computed, VI typically works well [Blei et al., 2003, Beal and Ghahramani, 2006, Hoffman et al., 2013]. However, when variable relationships are non-conjugate, a direct application of VI to the model would usually be intractable, because the related expectation terms either do not have any analytic form (for continuous variables), or require exponential complexities to calculate (for discrete ones). VI becomes even harder to use on models with complex structures created for large-scale, high-dimensional data, such as natural images, text documents, and community networks that are ubiquitous in our digital world.

In this dissertation, we propose solutions to these problems. Our goal is *not* to come up with one single algorithm that fits all the possible situations well, but to provide different strategies depending on the amount of control or information we have on the probabilistic models. Concretely, three advanced frameworks for variational inference are created.

Chapter 2[*] considers the most flexible case, where we are given the design control of not only the variational algorithm, but also the probabilistic model itself. In settings like this, we are able to make the best of both worlds, and create joint modeling-inference frameworks that capture the data characteristics in an organic manner.

The data observations we would focus on in this chapter are the pictures that could be taken by standard cameras in real-world environments, or so-called natural images. We propose an interpretable generative model where the hundreds of thousands of pixels per image are split into small patches. By doing this, the high-dimensional data get decomposed into local regions of only a few dozens of dimensions. We use zero-mean Gaussian mixture models (GMMs) to effectively capture their densities, as in Zoran and Weiss [2012].

Then the central problem to deal with is how to capture the dependencies between patches. Albeit our solution is an organized whole, for explanation purpose we dissemble it into two

[*]Contents of this chapter are mainly based on our work "From Patches to Images: A Nonparametric Generative Model" published in ICML 2017. Authors are Geng Ji, Michael C. Hughes, and Erik B. Sudderth.

separate parts, both of which involve the collaborative design of modeling and inference.

The first contribution is the creation of the random grid-location variable. During the generative process, we assume the image canvas is split by an unobserved grid into *non-overlapping* patches, but the alignment between them is set unknown. Then during inference, one needs to consider all the potential alignments, which essentially takes all the *overlapping* patches into account and thus enforces posterior dependencies. To balance between the prior and likelihood terms in the inference updates, we find that patch-level variables need to be jointly modeled and inferred with their corresponding grid alignment. The resulting structured variational objective on such a model automatically recovers the popular, hand-crafted expected patch log-likelihood method (EPLL, Zoran and Weiss [2011]), when combined with the GMM prior. More generally, this modeling-inference compound provides a highly principled justification for many image restoration pipelines that break images into overlapping patches, restore each of them, and aggregate to produce the final output [Elad and Aharon, 2006, Dabov et al., 2008, Mairal et al., 2009, Zoran and Weiss, 2011].

The other contribution is that we enhance the universal mixture weights across all patches via the hierarchical Dirichlet process (HDP, Teh et al. [2006]), in order to capture the self-similarities and repetitions that are ubiquitous in natural images [Jégou et al., 2009, Shaham et al., 2019], just like the word burstiness in text documents [Doyle and Elkan, 2009]. Similar to Hughes and Sudderth [2013], we introduce tractable lower bounds to overcome the non-conjugacy of HDP. More importantly, the local truncation setup allows us to create image-specific clusters during inference, which helps boost image denoising and inpainting accuracy on all the benchmark datasets tested.

Chapter 3[†] focuses on the case where the probabilistic models are already given and fixed. Then the task is to develop VI algorithms tailored to them that fulfill requirements such as high accuracy, high efficiency, and scalability to large datasets. We show that by making proper use of the probabilistic relations and structural dependencies between variables of the given model, it is possible to achieve all these goals jointly.

Specifically, the models of interests in this chapter are the hierarchical noisy-OR Bayesian networks. Just like the logical-OR operation, the noisy-OR conditionals assume the activation of a binary variable is independently influenced by its active parents [Horvitz et al., 1988]. This property has been widely used in the research of medical diagnosis, where bipartite noisy-OR graphs have been created to capture the relations between latent diseases in the top layer, and observed symptoms at the bottom [Shwe et al., 1991]. More recently, Liu et al. [2016] learns *hierarchical* noisy-OR topic models for text documents, in which the hidden directed acyclic graphs at the top indicate themes in different levels of abstraction, and the leaf-node observations correspond to the presence and absence of words in the vocabulary. Similar systems have also been used by IT companies like Google to analyze the semantic content of massive text datasets, as mentioned in Murphy [2012, Section 26.5.4].

To develop VI algorithms for topic models like this, the first challenge is to overcome the non-conjugacy caused by the noisy-OR conditionals. Inspired by Jaakkola and Jordan [1999], we construct tractable bounds by leveraging the log concavity of noisy-OR, and derive variational updates for arbitrary directed acyclic graphs, rather than bipartite networks only. Additionally, we further prove that the optimization w.r.t the auxiliary variables introduced to the variational objective is a concave problem, whose global optimum could be quickly found through the fixed-point iterations by Jaakkola and Jordan [1999].

[†]Contents of this chapter are mainly based on our work "Variational Training for Large-scale Noisy-OR Bayesian Networks" published in UAI 2019. Authors are Geng Ji, Dehua Cheng, Huazhong Ning, Changhe Yuan, Hanning Zhou, Liang Xiong, and Erik B. Sudderth.

Another important contribution for inference is the construction of *local* models for each document. The motivation is that while the vocabulary size and the number of topics for the entire model could be huge, each document usually just activates a very small portion of them. We make use of this property by further constructing a constrained family of variational bounds where only the ancestors of the sparse observations of each document need to be considered. We show that this treatment improves the running time by orders of magnitude, with negligible change in prediction accuracy.

Finally, we develop an efficient stochastic variational inference (SVI, Hoffman et al. [2013]) framework that learns more than one million noisy-OR edge weights on datasets containing hundreds of thousands of documents. It naturally integrates the inference strategies discussed above as the variational expectation step, and computes fast gradients for stochastic optimization w.r.t edges both inside and outside the local models.

Chapter 4[‡] considers the last situation where we need to build one single variational inference algorithm to directly work on a broad family of probabilistic models well. While this may sound a little over-demanding, with the increasing popularity of probabilistic programming languages (PPLs), there has been huge practical demand of general-purpose inference frameworks that allow new models to be easily setup and quickly tested. As a successful example, automatic differentiation variational inference (ADVI, Kucukelbir et al. [2017]) has provided such a systematic tool for models with reparameterizable continuous variables [Kingma and Welling, 2014, Kucukelbir et al., 2015].

We develop another Monte Carlo variational inference strategy that is is applicable to all models with *discrete* variables. Specifically, we use sampling to approximate expectations needed for optimal variational parameter updates. Compared to numerical evaluations [Jor-

[‡]Contents of this chapter are mainly based on our work "Effective Monte Carlo Variational Inference for Probabilistic Programs with Binary Variables" currently under review for AISTATS 2020. Authors are Geng Ji and Erik B. Sudderth.

dan et al., 1999, Blei et al., 2017], the complexity of our efficient Monte Carlo treatment is just linear in the number of samples even for models with high-order dependencies. Compared to REINFORCE-style stochastic gradient updates [Paisley et al., 2012b, Wingate and Weber, 2013, Ranganath et al., 2014], our coordinate-ascent updates have lower variance and converge much faster. Compared to auxiliary-variable bounds crafted for each individual non-conjugate relation [Albert and Chib, 1993, Jaakkola and Jordan, 1999, Polson et al., 2013], our algorithm is simpler to derive and may be easily integrated into probabilistic programming languages for broader use. By avoiding auxiliary variables, we also tighten likelihood bounds and increase robustness to local optima. Extensive experiments on real-world models of images, text, and networks illustrate these appealing advantages.

Chapter 5 concludes the dissertation. We summarize the entire work, highlight key contributions, and point out potential future research directions.

Chapter 2

Structured VI for Nonparametric Mixture Models of Natural Images

In this chapter, we propose a hierarchical generative model that captures the self-similar structure of image regions as well as how this structure is shared across image collections. Our model is based on a novel, variational interpretation of the popular expected patch log-likelihood (EPLL, Zoran and Weiss [2011]) method as a model for randomly positioned grids of image patches. While previous EPLL methods modeled image patches with finite Gaussian mixtures [Zoran and Weiss, 2012], we use nonparametric Dirichlet process (DP) mixtures to create models whose complexity grows as additional images are observed. An extension based on the hierarchical DP [Teh et al., 2006] then captures repetitive and self-similar structure via image-specific variations in cluster frequencies. We derive a structured variational inference algorithm that adaptively creates new patch clusters to more accurately model novel image textures. Our denoising performance on standard benchmarks is superior to EPLL and comparable to the state-of-the-art, and we provide novel statistical justifications for common image processing heuristics. We also show accurate image inpainting results. This chapter was previously published as [Ji et al., 2017].

2.1 Introduction

Models of the statistical structure of natural images play a key role in computer vision and image processing [Srivastava et al., 2003]. Due to the high dimensionality of the images captured by modern cameras, a rich research literature instead models the statistics of small image patches. For example, the K-SVD method [Elad and Aharon, 2006] generalizes K-means clustering to learn a dictionary for sparse coding of image patches. The state-of-the-art *learned simultaneous sparse coding* (LSSC, Mairal et al. [2009]) and *block matching and 3D filtering* (BM3D, Dabov et al. [2008]) methods integrate clustering, dictionary learning, and denoising to extract information directly from a single corrupted image. Alternatively, the accurate *expected patch log-likelihood* (EPLL, Zoran and Weiss [2011]) method maximizes the log-likelihood of overlapping image patches under a finite Gaussian mixture model learned from uncorrupted natural images.

We show that with minor modifications, the objective function underlying EPLL is equivalent to a variational log-likelihood bound for a novel generative model of whole images. Our model coherently captures overlapping image patches via a randomly positioned spatial grid. By deriving a rigorous variational bound, we then develop improved nonparametric models of natural image statistics using the *hierarchical Dirichlet process* (HDP, Teh et al. [2006]). In particular, DP mixtures allow an appropriate model complexity to be inferred from data, while the hierarchical DP captures the patch self-similarities and repetitions that are ubiquitous in natural images [Jégou et al., 2009, Kong and Fowlkes, 2018, Shaham et al., 2019]. Unlike previous whole-image generative models such as *fields of experts* (FoE, Roth and Black [2005]), which uses a single set of Markov random field parameters to model all images, our HDP model learns image-specific clusters to accurately model distinctive textures. Coupled with a scalable structured variational inference algorithm, we improve on the excellent denoising accuracy of the LSSC and BM3D algorithms, while providing a Bayesian

nonparametric model with a broader range of potential applications.

2.2 Expected Patch Log-likelihood

Our approach is derived from models of small (8×8 pixel) patches of a large natural image x. Let P_i be a binary indicator matrix that extracts the $G = 8^2$ pixels $P_i x \in \mathbb{R}^G$ in patch i. To reduce sensitivity to lighting variations, a *contrast normalizing* transform is applied to remove the mean (or "DC component") of the pixel intensities in each patch:

$$v_i = P_i x - \tfrac{1}{G}\mathbf{1}^T P_i x = B P_i x, \tag{2.1}$$

for a "zero-centering" matrix B. Zoran and Weiss [2012] show that a finite mixture of K zero-mean Gaussians,

$$p(v_i) = \textstyle\sum_{k=1}^{K} \pi_k \mathrm{Norm}(v_i \mid 0, \Lambda_k^{-1}), \tag{2.2}$$

is superior to many classic image models in terms of predictive likelihood and patch denoising performance.

The widely-used EPLL image restoration framework measures the quality of a reconstruction by the expected patch log-likelihood, "assuming a patch location in the image is chosen uniformly at random" [Zoran and Weiss, 2011]. Given a corrupted image y, EPLL estimates a clean image x by minimizing the objective:

$$\min_x \ \frac{\lambda}{2}\|x - y\|^2 - \textstyle\sum_i \log p(B P_i x). \tag{2.3}$$

Here, the sum ranges over all *overlapping*, completely visible (uncropped) image patches. The constant λ is determined by the noise level of the corrupted image y.

Direct optimization of Equation (2.3) is challenging, so inspired by *half quadratic split-*

ting [Geman and Yang, 1995], the EPLL objective can be reformulated as follows:

$$\min_{x,\bar{v}} \ \frac{\lambda}{2}\|x-y\|^2 + \sum_i \frac{\kappa}{2}\|P_i x - \bar{v}_i\|^2 - \log p(B\bar{v}_i). \tag{2.4}$$

Each patch i is allocated an auxiliary variable $\bar{v}_i$, which (unlike the v_i variable in Equation (2.1)) includes an estimate of the mean patch intensity. This augmented objective leads to closed-form coordinate descent updates.

Gating. Assign each patch i to some cluster z_i:

$$z_i = \arg\max_k \ \pi_k \ \text{Normal}\big(BP_i x \mid 0, \Lambda_k^{-1} + \kappa I\big). \tag{2.5}$$

Filtering. Given an approximate clean image x and cluster assignments z, denoise patches via least squares:

$$\bar{v}_i = \Big(I + \kappa^{-1} B^T \Lambda_{z_i} B\Big)^{-1} P_i x. \tag{2.6}$$

Mixing. Given a fixed set of auxiliary patches $\bar{v}$ and the noisy image y, a denoised image x is estimated as

$$x = \Big(\lambda I + \kappa \sum_i P_i^T P_i\Big)^{-1} \Big(\lambda y + \kappa \sum_i P_i^T \bar{v}_i\Big). \tag{2.7}$$

Annealing. Optimal solutions of Equation (2.4) approach those of the EPLL objective in Equation (2.3) as $\kappa \to \infty$. EPLL denoising algorithms slowly increase κ via an annealing schedule that must be tuned for best performance.

Justification? Empirically, the intuitive EPLL objective is much more effective than baselines which use only a subset of non-overlapping patches, or average independently denoised

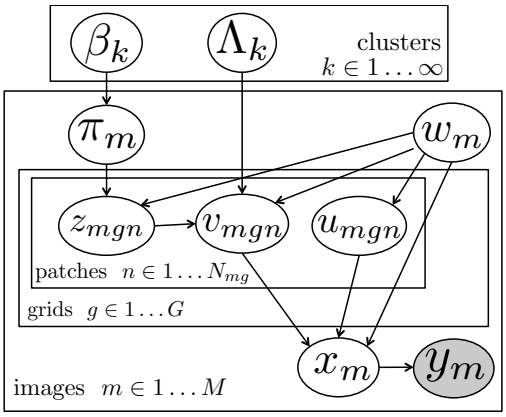


Figure 2.1: Directed graphical model for our HDP-Grid model of M natural images. Clean image x_m is generated via a randomly placed grid w_m of patches v_m generated by a hierarchical Gaussian mixture model. We observe corrupted images y_m.

patches [Zoran and Weiss, 2011]. But why should we optimize the expected *log*-likelihood, instead of the expected likelihood or another function of patch-specific likelihoods? And how can the EPLL heuristic be generalized to capture more complex statistics of natural images? This chapter answers these questions by linking EPLL to a rigorous, nonparametric generative model of whole images.

2.3 Mixture Models for Grids of Image Patches

We now develop the HDP-Grid generative model summarized in Figure 2.1, which uses randomly placed patch grids to formalize the EPLL objective, and hierarchical DP mixtures to capture image patch self-similarity.

2.3.1 Hierarchical Dirichlet Process Mixtures

The *hierarchical Dirichlet process* (HDP, Teh et al. [2006]) is a Bayesian nonparametric prior used to cluster groups of related data; we model natural images as groups of patches. The HDP shares visual structure, such as patches of grass or bricks, by sharing a common set of clusters (called *topics* in applications to text data) across images. In addition, the HDP

models image-specific variability by allowing each image to use this shared set of clusters with unique frequencies; grass might be abundant in one image but absent in another. Via the HDP, we can learn the proper number of hidden clusters from data, and discover new clusters as we collect new images with novel visual textures.

The HDP uses a stick-breaking construction to generate a corpus-wide vector

$$\pi_0 = [\pi_{01}, \pi_{02}, \ldots, \pi_{0k}, \ldots] \tag{2.8}$$

of frequencies for a countably infinite set of visual clusters:

$$\beta_k \sim \text{Beta}(1, \gamma), \quad \pi_{0k}(\beta) \triangleq \beta_k \prod_{\ell=1}^{k-1}(1 - \beta_\ell). \tag{2.9}$$

The HDP allocates each image m its own cluster frequencies π_m, where the vector π_0 determines the mean of a DP prior on the frequencies of shared clusters:

$$\pi_m \sim \text{DP}(\alpha\pi_0), \qquad \mathbb{E}[\pi_{mk}] = \pi_{0k}. \tag{2.10}$$

When the concentration parameter $\alpha < 1$, we capture the "burstiness" and self-similarity of natural image regions [Jégou et al., 2009] by placing most probability mass in π_m on a sparse subset of global clusters.

2.3.2 Image Generation via Random Grids

We sample pixels in image m via a randomly placed grid of patches. When each patch has G pixels, Figure 2.2 shows there are exactly G grid alignments for an image of arbitrary size. The alignment $w_m \in \{1, \ldots, G\}$ has a uniform prior:

$$w_m \sim \text{Categorical}(1/G, \ldots, 1/G). \tag{2.11}$$

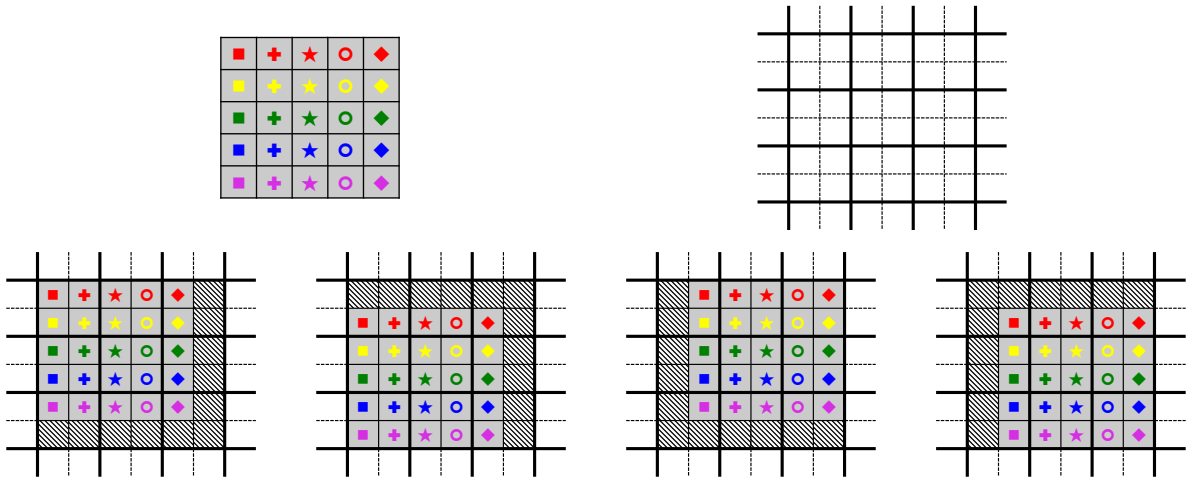


Figure 2.2: Generation of a complete image via a randomly positioned grid of non-overlapping patches. *Top left:* A 5×5 pixel image, where each pixel is identified by a distinct colored symbol. *Top right:* An infinite 2D grid of pixels, divided into 2×2 patches. *Bottom:* The four possible ways a 5×5 image may be generated from 2×2 patches. Shaded pixels are clipped by the image boundary (see Section 2.3.4).

Modeling multiple overlapping grids is crucial to capture real image statistics. As the true grid alignment for each image is uncertain, posterior inference will favor images that are likely under *all* possible w_m. Models based on a single, fixed grid produce severe artifacts at patch boundaries, as shown in Figure 2 of Zoran and Weiss [2011].

2.3.3 Patch Generation via Gaussian Mixtures

Gaussian mixtures provide excellent density models for natural image patches [Zoran and Weiss, 2012]. We associate clusters with zero-mean, full-covariance Gaussian distributions on patches with G pixels. We parameterize cluster k by a precision (inverse covariance) matrix $\Lambda_k \sim \text{Wishart}(\nu, W)$, whose conjugate Wishart prior has ν degrees of freedom and scale matrix W. Given that $w_m = g$, each of the N_{mg} patches v_{mgn} in grid g is sampled from an infinite mixture with image-specific cluster frequencies:

$$p(v_{mgn} \mid w_m = g) = \sum_{k=1}^{\infty} \pi_{mk} \text{Normal}(v_{mgn} \mid 0, \Lambda_k^{-1}). \tag{2.12}$$

Let $z_{mgn} \mid w_m = g \sim \text{Categorical}(\pi_m)$ denote the cluster that generates patch n. To account for the contrast normalization of Equation (2.1), the intensities in patch n are shifted by an independent, scalar "DC offset" u_{mgn}:

$$p(u_{mgn} \mid w_m = g) = \text{Normal}(u_{mgn} \mid r, s^2). \tag{2.13}$$

Finally, if $w_m \neq g$ so that grid g is unobserved, we sample $(z_{mgn}, v_{mgn}, u_{mgn})$ from some reference distribution independent of the HDP mixture model parameters.

2.3.4 From Patches to Corrupted Images

Given patches v_{mg} with offsets u_{mg} generated via grid $w_m = g$, we sample a whole "clean image" x_m as

$$\text{Normal}\Big(x_m \mid \sum_{n=1}^{N_{mg}} P_{mgn}^T \bar{v}_{mgn}, \delta^2 I\Big), \tag{2.14}$$

where $\bar{v}_{mgn} \triangleq C_{mgn} v_{mgn} + u_{mgn}$. Binary indicator matrices P_{mgn}, as in Section 2.2, stitch together patches in the chosen grid g. Image x_m is then generated by adding independent Gaussian noise with small variance δ^2. Most patches in the chosen grid will be fully observed in x_m, but as illustrated in Figure 2.2, some may be clipped by the image boundary. Indicator matrices C_{mgn} are defined so $C_{mgn} v_{mgn} + u_{mgn}$ is a vector containing the observed pixels from patch n.

For image restoration tasks, the observed image y_m is a corrupted version of some clean image x_m that we would like to estimate. Models of natural image statistics are commonly validated on the problem of image denoising, where x_m is polluted by additive white Gaussian

noise:

$$p(y_m \mid x_m) = \text{Normal}(y_m \mid x_m, \sigma^2 I). \tag{2.15}$$

The variance $\sigma^2 \gg \delta^2$ indicates the noise level. We also validate our model on image inpainting problems [Bertalmio et al., 2000], where some pixels are observed without noise but others are completely missing. By replacing Equation (2.15) with other linear likelihood models, our novel generative model for natural images may be easily applied to other tasks including image deblurring [Zoran and Weiss, 2011], image super resolution [Yang and Huang, 2010], and color image demosaicing [Mairal et al., 2009].

2.4 Variational Inference

We now develop scalable learning algorithms for our nonparametric, grid-based image model. We first examine a baseline *DP Grid* model in which the same cluster frequencies π_0 are shared by all images. Our full *HDP Grid* model then learns image-specific cluster frequencies π_m, and instantiates new clusters to model unique visual textures.

2.4.1 DP Grid: Variational Inference

Our goal is to infer the DP Grid model parameters that best explain observed images which may be clean (x_m) or corrupted by noise (y_m). The DP Grid model uses the same cluster probabilities π_0, generated from stick-breaking weights β as in Equation (2.9), for all images.

Learning from clean images

Given a training set $\mathcal{D}$ of *uncorrupted* images $x_1, \dots x_M$, we estimate the posterior distribution $p(\beta, \Lambda, w, \Psi^{\text{patch}} \mid x)$ for our global mixture model parameters β and Λ, grid assignment

indicators w_m, and patch-level latent variables $\Psi_m^{\text{patch}} = \{u_m, v_m, z_m\}$.

Exact posterior inference is intractable, so we instead find an approximate posterior $q(\cdot) = q(\beta, \Lambda, w, \Psi^{\text{patch}})$ minimizing the KL divergence [Wainwright and Jordan, 2008] from the true posterior $p(\cdot|x)$. Equivalently, our variational method maximizes the following objective $\mathcal{L}$:

$$\max_{q \in \mathcal{Q}} \mathcal{L}(q, x) = \max_{q \in \mathcal{Q}} \mathbb{E}_q \left[\log \frac{p(x, \cdot)}{q(\cdot)} \right] \leq \log p(x). \tag{2.16}$$

We constrain the solution of our optimization to come from a tractable family of *structured* mean-field distributions $\mathcal{Q}$, parameterized by free parameters. Unlike naïve mean-field methods which assume complete posterior independence, our structured mean-field approximation is more accurate and includes dependencies between some latent variables:

$$q(\cdot) = \prod_{k=1}^{\infty} q(\Lambda_k) q(\beta_k) \cdot \prod_{m=1}^{M} q(w_m) q(\Psi_m^{\text{patch}} \mid w_m). \tag{2.17}$$

As in Hughes and Sudderth [2013], this approximate posterior family contains *infinitely* many clusters, just like the true posterior. Rather than applying a fixed truncation to the stick-breaking prior [Blei and Jordan, 2006], we dynamically truncate the patch assignment distributions $q(z)$ to only use the first K clusters to explain the M observed images. Clusters with indices $k > K$ then have factors $q(\Lambda_k)$ set to the prior, and need not be explicitly represented.

Global mixture model

The global cluster weights β and precision matrices Λ have standard exponential family forms (free parameters are marked by hats):

$$q(\Lambda_k) = \text{Wishart}\big(\hat{\nu}_k, \hat{W}_k\big), \quad q(\beta_k) = \text{Beta}\big(\hat{\rho}_k \hat{\omega}_k, (1 - \hat{\rho}_k)\hat{\omega}_k\big). \tag{2.18}$$

Here $\hat{\rho}_k = \mathbb{E}_q[\beta_k]$, and $\hat{\omega}_k$ controls the variance of $q(\beta_k)$.

Image-specific alignment

For natural images, all grid alignments are typically of similar quality, so we fix a *uniform* alignment posterior $q(w_m) = \text{Categorical}\big(\frac{1}{G}, \ldots, \frac{1}{G}\big)$. This simplifies many updates while still avoiding artifacts that would arise from a single, non-overlapping patch grid.

Patch-specific factors

The patch-specific variables Ψ^{patch} have *structured* posteriors, conditioned on the value of the grid indicator w_m for the current image:

$$\begin{aligned} q(z_{mgn} \mid w_m = g) &= \text{Categorical}\big(\hat{r}_{mgn1}, ..., \hat{r}_{mgnK}\big), \\ q(u_{mgn} \mid w_m = g) &= \text{Normal}\big(\hat{u}_{mgn}, \hat{\phi}^u_{mgn}\big), \\ q(v_{mgn} \mid w_m = g, z_{mgn} = k) &= \text{Normal}\big(\hat{v}_{mgnk}, \hat{\phi}^v_{mgnk}\big). \end{aligned} \tag{2.19}$$

Below, we let $\mathbb{E}_q[\cdot]$ denote the *conditional* expectation with respect to the variational distribution q, given w_m.

Learning

Given clean images x, we perform coodinate ascent on the objective $\mathcal{L}$, alternatively updating one factor among $q(\beta)q(\Lambda)q(w)q(\Psi^{\text{patch}})$. Most updates have closed forms due to the exponential families defining $\mathcal{Q}$, as presented in Appendix 2.A. As one intuitive example,

consider the update for the cluster precision matrix posterior $q(\Lambda_k \mid \hat{\nu}_k, \hat{W}_k)$:

$$\hat{\nu}_k = \nu + \underbrace{\frac{1}{G}\sum_{m=1}^{M}\sum_{g=1}^{G}\sum_{n=1}^{N_{mg}} \hat{r}_{mgnk}}_{N_k}, \quad \hat{W}_k = W + \underbrace{\frac{1}{G}\sum_{m=1}^{M}\sum_{g=1}^{G}\sum_{n=1}^{N_{mg}} \mathbb{E}_q\left[\mathbb{1}_k(z_{mgn}) v_{mgn} v_{mgn}^T\right]}_{S_k}. \quad (2.20)$$

Statistic $N_k(\hat{r})$ counts patches assigned to cluster k, while $S_k(\hat{r}, \hat{v}, \hat{\phi}^v)$ aggregates second moments. These updates follow the standard form of prior parameter plus expected sufficient statistic, except the statistics are averaged (*not* simply added) across the G grid alignments.

2.4.2 Image Denoising and Connections to EPLL

Given a corrupted image y_m, we seek to compute the posterior $p(x_m \mid y_m, \mathcal{D})$, where we condition on the training set $\mathcal{D}$. Our variational posterior family Q now includes an additional factor for the unobserved, "clean" image x_m:

$$q(x_m) = \text{Normal}\big(x_m \mid \hat{x}_m, \hat{\phi}_m^x\big). \quad (2.21)$$

The variational inference objective becomes

$$\max_{q \in \mathcal{Q}} \mathbb{E}_q\left[\log \frac{p(\mathcal{D}, y_m, x_m, \cdot)}{q(x_m, \cdot)}\right] \leq \log p(y_m, \mathcal{D}), \quad (2.22)$$

and the coordinate ascent update for $q(x_m)$ equals

$$\hat{x}_m = \hat{\phi}_m^x\Big(\frac{y_m}{\sigma^2} + \frac{h_m}{\delta^2}\Big), \quad \hat{\phi}_m^x = \frac{\delta^2\sigma^2}{\delta^2 + \sigma^2} I. \quad (2.23)$$

The updated covariance is diagonal, improving computational efficiency. The mean depends on the average image vector across all patches in all grids, denoted by h_m:

$$h_m \triangleq \frac{1}{G} \sum_{g=1}^{G} \sum_{n=1}^{N_{mg}} P_{mgn}^T (C_{mgn} \mathbb{E}_q[v_{mgn}] + \hat{u}_{mgn}). \tag{2.24}$$

Note that the update for $\hat{x}_m$ in Equation (2.23) is similar to the EPLL update in Equation (2.7), except that some terms involving projection matrices become constants because we account for partially observed patches. Modeling partial patches is necessary to produce a valid likelihood bound in Equation (2.22).

In fact, as we show below all three terms in the EPLL objective in Equation (2.4) are very similar to our proposed minimization objective function $-\mathcal{L}$, up to a scale factor of G. Of course, a key difference is that our objective seeks full posteriors rather than point estimates, and enables the HDP model of multiple images detailed in Section 2.4.3.

EPLL Term 1

When we set $\lambda \triangleq \frac{G}{\sigma^2}$, the first term of the EPLL objective in Equation (2.4) becomes

$$G \cdot \tfrac{1}{2\sigma^2} (x - y)^T (x - y). \tag{2.25}$$

Similarly, suppressing the subscript m denoting the image for simplicity, $\mathbb{E}_q[-\log p(y|x)]$ in our $-\mathcal{L}$ simplifies as

$$\tfrac{1}{2\sigma^2} \mathbb{E}_q[(x - y)^T (x - y)]. \tag{2.26}$$

EPLL Term 2

Taking the second term in Equation (2.4) and substituting $\kappa = 1/\delta^2$, we have:

$$\tfrac{1}{2\delta^2} \textstyle\sum_i (P_i x - \bar{v}_i)^T (P_i x - \bar{v}_i). \tag{2.27}$$

The corresponding term $\mathbb{E}_q[-\log p(x \mid w, u, v)]$ in our objective $-\mathcal{L}$ can be written similarly up to a scaling by G:

$$\frac{1}{G}\frac{1}{2\delta^2} \sum_{g=1}^{G} \sum_{n=1}^{N_g} \mathbb{E}_q\Big[(P_{gn} x - \bar{v}_{gn})^T (P_{gn} x - \bar{v}_{gn})\Big]. \tag{2.28}$$

EPLL Term 3

The third EPLL term assumes zero-centered patches $B\bar{v}_i$ are drawn from Gaussian mixtures:

$$-\textstyle\sum_i \log p(B\bar{v}_i \mid \pi_0, \Lambda). \tag{2.29}$$

Similarly, in our minimization objective $-\mathcal{L}$ we draw v_{gn} from a DP mixture model. Explicitly including the cluster assignment z_{gn}, $\mathbb{E}_q[-\log p(v, z \mid w)]$ equals

$$-\frac{1}{G} \sum_{g=1}^{G} \sum_{n=1}^{N_g} \mathbb{E}_q[\log p(v_{gn}, z_{gn} \mid \pi_0, \Lambda)]. \tag{2.30}$$

EPLL is similar, but maximizes assignments (Equation (2.5)) rather than computing posterior assignment probabilities.

2.4.3 HDP Grid: Variational Inference

Image-specific frequencies

The DP model above, and the parametric EPLL objective it generalizes, assume the same cluster frequency vector π_0 for each image m. Our HDP Grid model allows image-specific frequencies π_m to be learned from data, via the hierarchical regularization of the HDP prior [Teh et al., 2006]. Our approximate posterior family $\mathcal{Q}$ now has the following HDP-specific factors:

$$q(\beta) = \prod_{k=1}^{\infty} \text{Beta}\left(\beta_k \mid \hat{\rho}_k\hat{\omega}_k, (1-\hat{\rho}_k)\hat{\omega}_k\right),$$
$$q([\pi_{m1}, \ldots, \pi_{mK}, \pi_{m>K}]) = \text{Dirichlet}(\hat{\theta}_{m1}, \ldots, \hat{\theta}_{mK}, \hat{\theta}_{m>K}). \tag{2.31}$$

This approximate posterior represents infinitely many clusters via a finite partition of π_m into $K+1$ terms: one for each of the K active clusters, and a remainder term at index $>K$ that aggregates the mass of all inactive clusters. The free parameter $\hat{\theta}_m$ is also a vector of size $K+1$ whose last entry represents all inactive clusters. We follow Hughes et al. [2015] to obtain a closed-form update for $\hat{\theta}_m$, and gradient-based updates for $\hat{\rho}, \hat{\omega}$; see Appendix 2.B for details. We highlight that the $\hat{\theta}_m$ update naturally includes a $\frac{1}{G}$ rescaling of count sufficient statistics as in Equation (2.20). Other factors remain unchanged from the DP Grid model.

Image-specific clusters

Due to the heavy-tailed distribution of natural images [Ruderman, 1997], even with large training sets, test images may still contain unique textural patterns like the striped scarf in the Barbara image in Figure 2.3. Fortunately, our Bayesian nonparametric HDP Grid model provides a coherent way to capture such patterns by appending K' novel, image-specific clusters to the original K clusters learned from training images. These novel clusters lead

to more accurate posterior approximations $q \in \mathcal{Q}$ that better optimize our objective $\mathcal{L}$.

We initialize inference by creating $K' = 100$ image-specific clusters with the `k-means++` algorithm [Arthur and Vassilvitskii, 2007], which minimizes the cost function

$$\mathcal{J}(z', \Lambda') = \sum_i \sum_{k=1}^{K'} \mathbb{1}_k(z'_i) \mathrm{D}(\tilde{v}_i \tilde{v}_i^T, \Lambda'_k), \tag{2.32}$$

where the first sum is over the set of fully-observed patches within the image. The function D is the Bregman divergence associated with our zero-mean Gaussian likelihood [Banerjee et al., 2005], and $\tilde{v}_i = BP_i y$ is a zero-centered patch. We initialize the algorithm by sampling K' diverse patches in a distance-biased fashion, and refine with 50 iterations of coordinate descent updates of z' and Λ'.

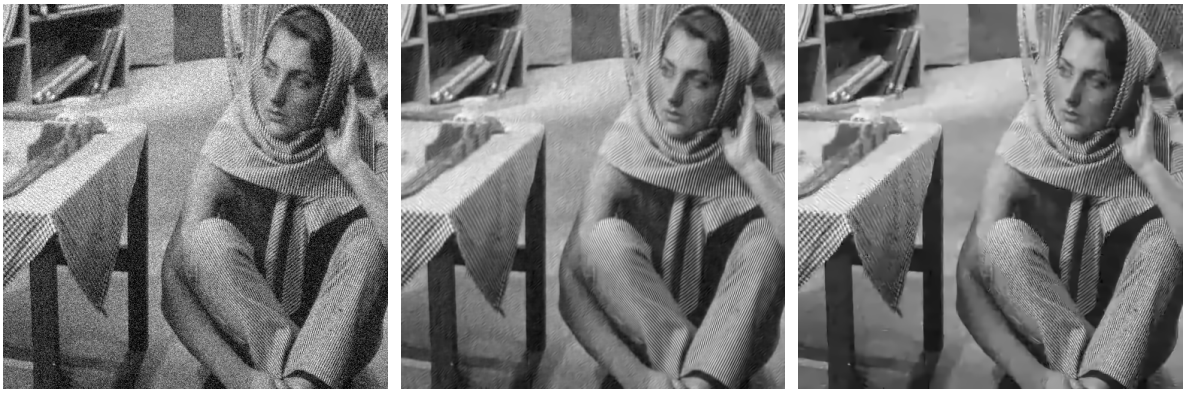
Then we expand the variational posterior $q(\Lambda)$ into $K + K'$ clusters. The first K indices are kept the same as training, and the last K' indices are set via Equation (2.20) using sufficient statistics N', S' derived from hard assignments z':

$$N'_{k'} \leftarrow \sum_i \mathbb{1}_{k'}(z'_i), \quad S'_{k'} \leftarrow \left[\sum_i \mathbb{1}_{k'}(z'_i) \tilde{v}_i \tilde{v}_i^T - N_{k'} \sigma^2 I \right]_+. \tag{2.33}$$

Here, following Portilla et al. [2003] and Kivinen et al. [2007], $S'_{k'}$ estimates the *clean* data statistic $S_{k'}$ by subtracting the expected noise covariance. The $[\cdot]_+$ operator thresholds any negative eigenvalues to zero.

Similarly, the other global variational factor $q(\beta)$ is also expanded to $K + K'$ clusters via sufficient statistics N' and counts of cluster usage from training data. Given $\{\beta, \Lambda\}_{k=1}^{K+K'}$, each factor in q may then be updated in turn to maximize the variational objective $\mathcal{L}$. See Appendix 2.B for details.

Finally, while we initialize K' to a large number to avoid local optima, this may lead to



Noisy: 20.19 dB | iDP: 29.41 dB | EPLL: 28.65 dB

eDP: 29.01 dB | HDP: **30.15** dB | HDP: new clusters

Figure 2.3: For an image with noise level $\sigma = 25$, the HDP improves denoising performance by leveraging both internal clusters (e.g., scarf and tablecloth) and external clusters (e.g., floor and table legs). The bottom right image colors the pixels assigned to each of 9 internal HDP clusters. **Best viewed electronically.**

extraneous clusters. We thus delete new clusters that our sparsity-biased variational updates do not assign to any patch. In the Barbara image in Figure 2.3, this leaves 9 image-specific clusters. Deletion improves model interpretability and algorithm speed, because costs scale linearly with the number of instantiated clusters.

2.5 Experiments

Following EPLL, we train our HDP-Grid model using 400 clean training and validation images from the Berkeley segmentation dataset (BSDS, Martin et al. [2001]). We fix $\delta = 0.5/255$

eDP: 32.47 dB HDP: **32.65** dB

Figure 2.4: By capturing self-similar patches in the "house" image, our HDP model reduces artifacts in smooth regions such as the sky, roof, and walls. Input noise level $\sigma = 25$ (PSNR=20.21 dB).

to account for the quantization of image intensities to 8-bit integers. Observed DC offsets u provide maximum likelihood estimates of the mean r and variance s^2 in Equation (2.13). Similarly, we compute empirical covariance matrices for patches in the same image segments to estimate hyperparameters W and ν in Equation (2.20). Using variational learning algorithms that adapt the number of clusters to the observed data [Hughes and Sudderth, 2013], we discover $K = 449$ clusters for the DP-Grid model, which we use to initialize our HDP model. We set our annealing schedule for κ to match that used by the public EPLL code.

Image denoising methods are often divided into two types Zontak and Irani [2011]: *external* methods (like EPLL) that learn all parameters from a training database of clean images, and *internal* methods that denoise patches using other patches of the single noisy image. For example, the K-SVD Elad and Aharon [2006] has an external variant that uses a dictionary learned from clean images, and an internal variant that learns its dictionary from the noisy image. A major contribution of our work is to show that the hierarchical DP leads to a

Table 2.1: Average PSNR values on benchmark datasets of classic-12 (top) and BSDS-68 (bottom). Larger values indicate better denoising. Methods are highlighted if they are indistinguishable with 95% confidence, according to a Wilcoxon signed-rank test on the fraction of images where one method outperforms another. For all noise levels, the patch size of BM3D is fixed to 8×8 and LSSC is fixed to 9×9.

σ	iDP	EPLL	eDP	HDP	FoE	eKSVD	iKSVD	BM3D	LSSC
10	33.66	33.68	33.77	**33.99**	33.11	33.45	33.62	33.98	**34.05**
25	29.02	29.39	29.47	**29.68**	28.32	28.89	29.11	**29.73**	**29.74**
50	25.44	26.22	**26.28**	**26.42**	24.69	25.44	25.64	**26.55**	**26.43**
10	33.10	33.37	33.42	**33.47**	32.69	33.06	33.08	33.26	33.45
25	28.33	28.72	28.76	**28.82**	27.76	28.28	28.28	28.55	28.70
50	25.10	25.72	25.75	**25.83**	24.48	25.17	25.17	25.59	25.50

Table 2.2: Average SSIM values on benchmark datasets of classic-12 (top) and BSDS-68 (bottom). Settings are the same as in Table 2.1.

σ	iDP	EPLL	eDP	HDP	FoE	eKSVD	iKSVD	BM3D	LSSC
10	0.9118	0.9136	**0.9143**	**0.9169**	0.8962	0.9084	0.9111	0.9168	**0.9185**
25	0.8189	0.8286	0.8299	**0.8337**	0.8018	0.8082	0.8131	**0.8357**	**0.8359**
50	0.6962	**0.7301**	**0.7316**	**0.7366**	0.6885	0.6926	0.6975	**0.7425**	**0.7390**
10	0.9119	0.9219	0.9224	**0.9230**	0.8971	0.9128	0.9135	0.9157	0.9206
25	0.7964	0.8090	0.8103	**0.8131**	0.7804	0.7859	0.7879	0.8010	0.8109
50	0.6636	0.6870	0.6880	**0.6962**	0.6585	0.6544	0.6539	0.6840	0.6885

principled hybrid of internal and external methods, in which cues from clean and noisy images are automatically combined in an adaptive way.

2.5.1 Image Denoising

We test our algorithm on 12 "classic" images used in many previous denoising papers [Mairal et al., 2009, Zoran and Weiss, 2011], as well as the 68 BSDS test images used by Roth and Black [2005], Zoran and Weiss [2011]. We evaluate the denoising performance by the *peak signal-to-noise ratio* (PSNR), a logarithmic transform of the *mean squared error* (MSE)

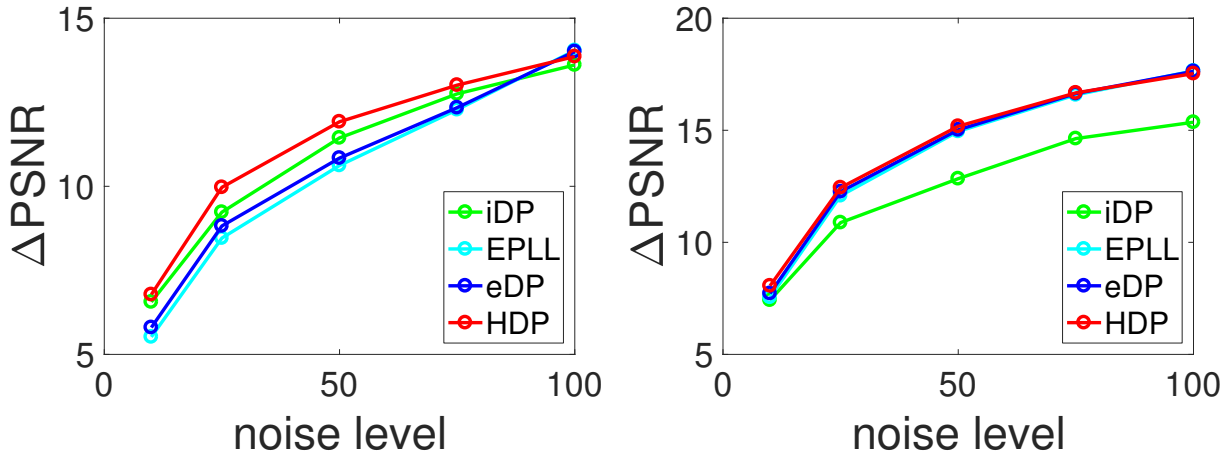


Figure 2.5: Denoising performance of grid-based models on the Barbara image of Figure 2.3 (left) and the house image of Figure 2.4 (right), as a function of the noise standard deviation. For both images and all noise levels, the HDP model is superior to baselines that solely use external (eDP) or internal (iDP) training, in terms of PSNR improvement relative to the noisy input image. When the image is extremely noisy ($\sigma = 100$), internal clusters are of poor quality, and the HDP and eDP models are comparable.

between images with normalized intensities,

$$\text{PSNR} \triangleq -20 \log_{10} \text{MSE}. \tag{2.34}$$

We also evaluate the *structural similarity index* (SSIM, Wang et al. [2004]), which quantifies image quality degradation via changes in structure, luminance, and contrast.

Internal vs. external clusters

In result figures, we use *eDP* to refer to our DP-Grid model trained solely on external clean images and *HDP* to refer to the HDP-Grid model that also learns novel image-specific clusters. We also train an internal DP-Grid model, referred to as *iDP*, using only information from the noisy test image. The first four columns of Table 2.1 and Table 2.2 compare their average denoising performance, where EPLL can be viewed as a simplification of eDP. For all noise levels and datasets, the HDP model has superior performance. As shown in Figure 2.6, HDP is more accurate than EPLL and eDP for every single classic-12 image.

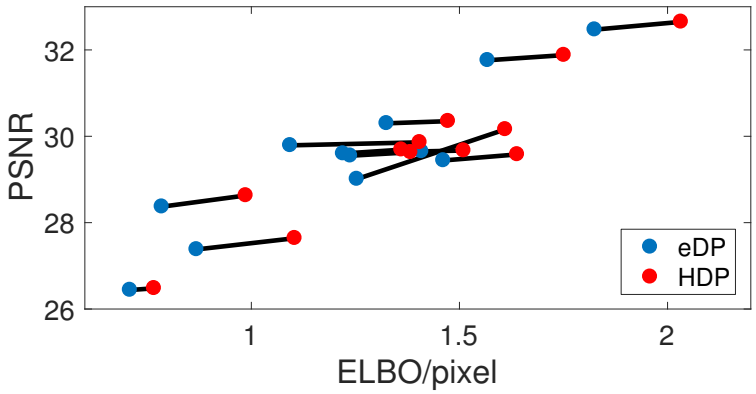


Figure 2.6: Clean-image evidence lower bound (ELBO) versus output PSNR ($\sigma = 25$) for 12 "classic" images. The horizontal axis plots $\log p(x_{\text{test}} \mid x_{\text{train}}) \approx \mathcal{L}(x_{\text{test}}, x_{\text{train}}) - \mathcal{L}(x_{\text{train}})$, divided by the number of pixels. Our HDP is uniformly superior to the eDP.

Also, the consistent gain in performance from EPLL to eDP demonstrates the benefits of Bayesian nonparametric learning of an appropriate model complexity (for EPLL, the number of clusters was arbitrarily fixed at $K = 200$).

Figure 2.3 further illustrates the complementary role of internal and external clusters for a single test image ("Barbara"). The *internal* iDP perfectly captures some unique textures like the striped clothing, but produces artifacts in smooth background regions. The *external* EPLL and eDP better represent smooth surfaces and contours, which are common in training data, but poorly recover striped textures.

As shown in Figure 2.5, while the relative accuracy of the eDP and iDP models varies depending on image statistics, the HDP model adaptively combines external and internal clusters for superior performance at all noise levels. By capturing the expected self-similarity of image patches, the HDP model also reduces artifacts in large regions with regular textures, such as the smoothly shaded areas of Figure 2.4.

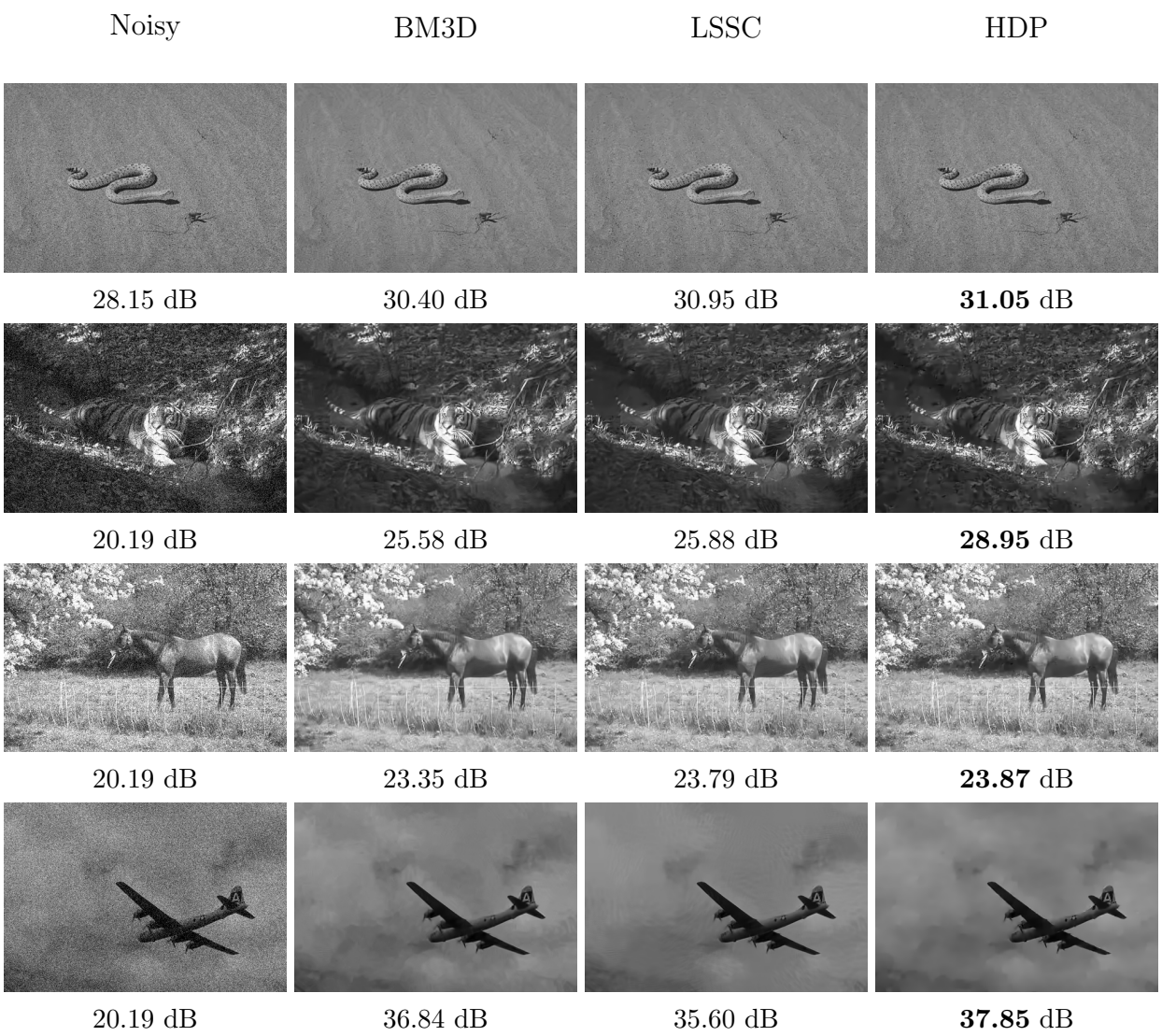


Figure 2.7: Comparison of image denoising methods on BSDS-68. Unlike our HDP model, the BM3D and LSSC methods learn solely from the noisy image and do not accurately capture some textures such as the sandy ground in *Row 1*, fallen leaves and tiger tail in *Row 2*, trees and grass in *Row 3*, and sky and clouds in *Row 4*. Noise level $\sigma = 10$ in *Row 1*, $\sigma = 25$ elsewhere. **Best viewed electronically.**

Computational speed

To denoise a 512×512 pixel image on a modern laptop, our Python code for eDP inference with $K = 449$ clusters takes about 12 min. The public EPLL Matlab code [Zoran and Weiss, 2011] with $K = 200$ clusters takes about 5 min. With equal numbers of clusters, the two methods have comparable runtimes. Our open-source Python code is available online at

`https://github.com/bnpy/hdp-grid-image-restoration`.

Learning image-specific clusters for the HDP model is more expensive: our non-optimized Python denoising code currently requires about 30 min. per image. Nearly all of the extra time is spent on the k-means++ initialization of Equation (2.32). We expect this can be sped up significantly by coding core routines in C/C++, parallelizing some sub-steps (possibly via GPUs), using fewer internal clusters (100 is often too many), or using faster initialization heuristics.

Performance

We compare our HDP model to other patch-based denoising methods in Table 2.1 and Table 2.2. On classic-12, where many top methods have been hand-tuned to perform well, our model is statistically indistinguishable from the best baselines. On the larger BSDS-68, our performance is superior to the state-of-the-art, showing the value of nonparametric learning from large image collections. See Figure 2.7 for examples. At higher noise levels ($\sigma = 50$), LSSC has modestly improved performance (0.2 dB in PSNR) when modeling 12×12 patches [Mairal et al., 2009]. HDP models of larger patches are a promising research area.

2.5.2 Image Inpainting

While many image processing systems are designed for just one problem, our generative model is useful for many tasks. For example, we can "inpaint" occluded image regions (like the red pixels in Figure 2.8) by modifying Equation (2.15) to let $\sigma^2 \to \infty$ for only those regions and setting $\sigma^2 = 0$ elsewhere. To process color images, we follow the approach of FoE and EPLL and convert to the YCbCr color space before independently inpainting each channel. While ground truth is unavailable for the classic image in Figure 2.8, our grid-based

HDP produces fewer visual artifacts than baselines.

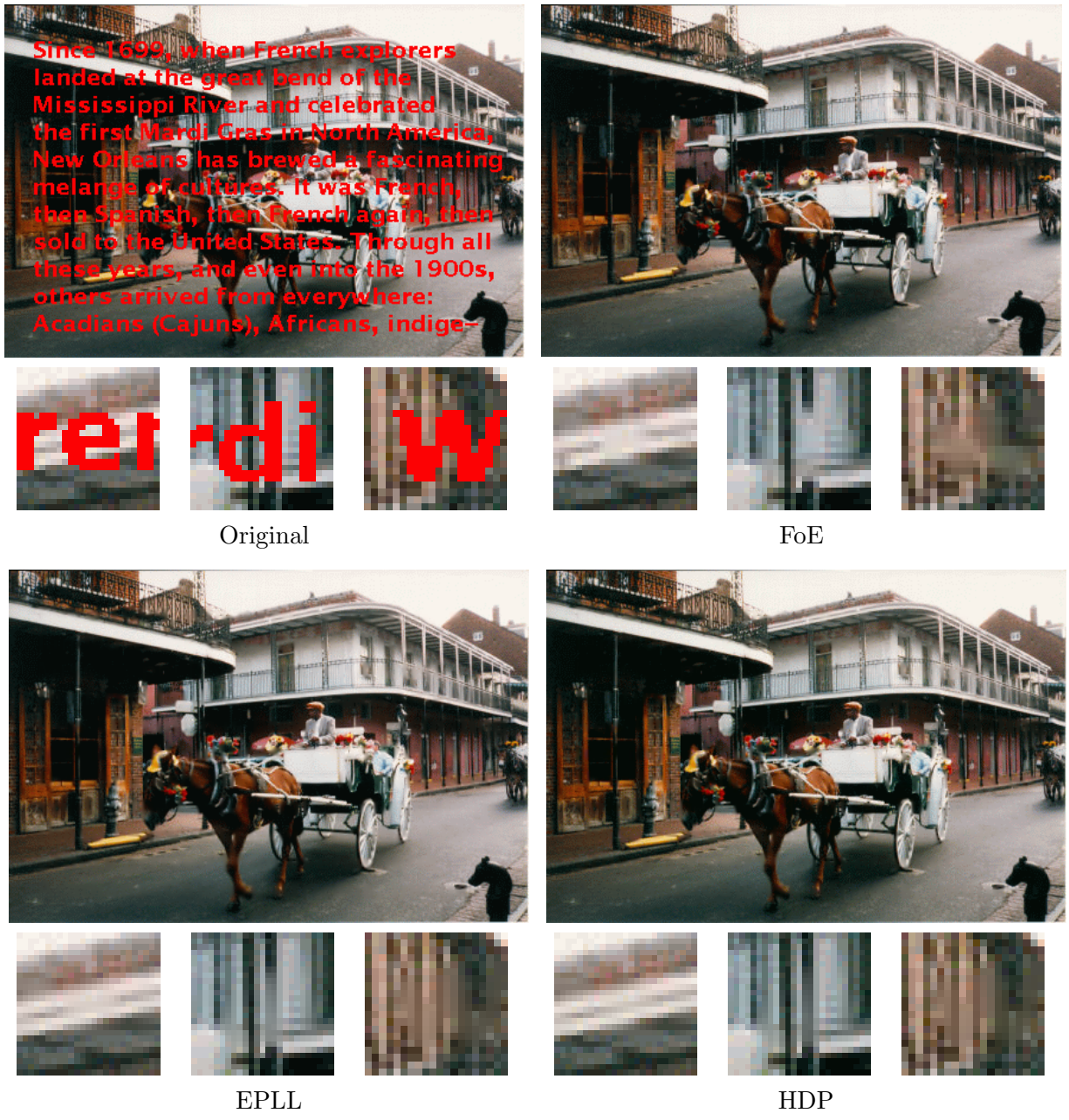


Figure 2.8: A qualitative comparison of image inpainting algorithms. As illustrated in the three close-up views, the HDP exploits patch self-similarity to better recover fine details.

2.6 Discussion

We have developed a coherent Bayesian nonparametric model that, via randomly positioned grids of image patches, provides a novel statistical foundation for the popular EPLL method. We show that HDP mixture models of visual textures can grow in complexity as additional images are observed and capture the self-similarity of natural images. Our HDP-grid image denoising and inpainting algorithms are competitive with the state-of-the-art, and our model is applicable to many other computer vision tasks.

2.A DP Grid: Variational Inference Details

As in the main text, our goal is to best explain many observed noisy images y_m with the DP Grid model. Specifically, we wish to use the variational distribution $q(\beta, \Lambda, x, w, \Psi^{\text{patch}})$ to estimate the posterior $p(\beta, \Lambda, x_m, w_m, \Psi_m^{\text{patch}} \mid y_m)$. In each subsection below, we look at a subset of random variables and discuss: (1) the chosen approximate posterior family, (2) useful expectations for computing terms of the variational objective $\mathcal{L}$, and (3) the coordinate ascent update equations that will improve $\mathcal{L}$.

2.A.1 Approximate Posterior for Global Random Variables

The DP mixture model has two global random variables which are shared across all images: the per-cluster stick-breaking frequencies β_k and the per-cluster precision matrix Λ_k. As shown in Equation (2.18), our chosen approximate posterior factors for these quantities have standard exponential-family forms. The Wishart approximate posterior $q(\Lambda_k)$ has a positive scalar $\hat{\nu}_k \in \mathbb{R}^+$ and a $G \times G$ positive definite matrix $\hat{W}_k$. The Beta posterior $q(\beta_k)$ has a positive scalar parameter $\rho_k \in [0, 1]$ which defines the mean of β_k, and another positive scalar $\omega_k \in \mathbb{R}^+$ which controls the variance.

Useful Expectations

Expectations for cluster-specific precision matrices. Under the chosen $q(\Lambda_k)$, we have the expectations:

$$\mathbb{E}_q[\Lambda_k] = \hat{\nu}_k \hat{W}_k^{-1}, \quad \mathbb{E}_q[\log|\Lambda_k|] = \sum_{g=1}^{G} \psi\Big(\frac{\hat{\nu}_k + 1 - g}{2}\Big) + G\log 2 - \log|\hat{W}_k|, \tag{2.35}$$

in which ψ stands for the derivative of the logarithm of gamma function, often called the *digamma* function.

Expectations for cluster frequencies. Under our chosen family for $q(\beta)$ we have closed-form expressions for key expectations of the cluster frequencies π_{0k} for active clusters $k \leq K$:

$$\mathbb{E}_q[\pi_{0k}] = \hat{\rho}_k \prod_{l=1}^{k-1}(1-\hat{\rho}_l), \quad \mathbb{E}_q[\log \pi_{0k}] = \sum_{l=1}^{k-1} \mathbb{E}_q[\log(1-\beta_l)] + \mathbb{E}_q[\log \beta_k]. \tag{2.36}$$

The remaining mass above some cluster index K is also known:

$$\mathbb{E}_q[\pi_{0>K}] = \prod_{\ell=1}^{K}(1-\hat{\rho}_\ell). \tag{2.37}$$

Closed-form expectations of direct functions of β_k:

$$\mathbb{E}_q[\log \beta_k] = \psi(\hat{\rho}_k\hat{\omega}_k) - \psi(\hat{\omega}_k), \quad \mathbb{E}_q[\log(1-\beta_k)] = \psi((1-\hat{\rho}_k)\hat{\omega}_k) - \psi(\hat{\omega}_k). \tag{2.38}$$

Coordinate Ascent Updates

Following Hughes and Sudderth [2013], we only explicitly compute posterior statistics for the K "active" clusters that have been assigned to at least one patch. All clusters with index $> K$ are by definition independent of the data. Thus, their posterior factors are simply equal to their priors [Hughes and Sudderth, 2013] and need not be instantiated.

Update for $q(\Lambda_k)$. The Wishart posterior for the corpus-wide cluster precision matrix Λ_k enjoys standard exponential family additive updates where the relevant sufficient statistics are N_k, an aggregated usage count, and S_k, an aggregated outer-product. As shown in Equation (2.20), these statistics are averaged across all G grid alignments.

Update for $q(\beta_k)$. For the DP-mixture, the optimal update to each cluster's stick-breaking weight $q(\beta_k)$ also has a standard closed form, as described in Hughes and Sudderth [2013]:

$$\hat{\rho}_k \hat{\omega}_k \leftarrow N_k + 1, \quad (1 - \hat{\rho}_k)\hat{\omega}_k \leftarrow N_{>k} + \gamma. \tag{2.39}$$

Here, the count $N_{>k}$ represents the aggregated statistic for all clusters with index larger than k: $N_{>k} = \sum_{\ell=k+1}^{K} N_k$.

2.A.2 Approximate Posterior for Patch Random Variables

The approximate posterior family for the patch-specific random variables u, v, z have been defined in Equation (2.19). We interpret the responsibility parameter $\hat{r}_{mgnk}$ as the posterior probability of assigning the n-th patch in grid g to the k-th cluster. The vector $\hat{r}_{mgn}$ must have K positive entries that sum to one. The posterior for scalar DC offset u has a simple Gaussian distribution with free mean and variance parameters. Similarly, the posterior for vector v has a Gaussian form with mean and covariance matrix.

Note that each of these factors conditions on the value of the grid indicator w_m for the current image m. This conditioning provides flexible posterior structures and elegant update equations not possible with naïve mean-field methods.

Useful Expectations

Under our structured approximate posterior, we have the following expectations:

$$\mathbb{E}_q[\mathbb{1}_k(z_{mgn})v_{mgn}] = \hat{r}_{mgnk}\hat{v}_{mgnk}, \quad \mathbb{E}_q[v_{mgn}] = \sum_{k=1}^{K} \hat{r}_{mgnk}\hat{v}_{mgnk}. \tag{2.40}$$

Similarly, we have the following outer-product expectations:

$$\mathbb{E}_q[\mathbb{1}_k(z_{gmn})v_{mgn}v_{mgn}^T] = \hat{r}_{mgnk}(\hat{v}_{mgnk}\hat{v}_{mgnk}^T + \hat{\phi}^v_{mgnk}), \tag{2.41}$$

and

$$\mathbb{E}_q[v_{mgn}v_{mgn}^T] = \sum_{k=1}^{K} \hat{r}_{mgnk}(\hat{v}_{mgnk}\hat{v}_{mgnk}^T + \hat{\phi}^v_{mgnk}). \tag{2.42}$$

Coordinate Ascent Updates

Updating $q(z \mid w)$. Within image m, we update the $n-$th patch inside the $g-$th grid by computing a scalar positive weight for each active cluster $k = 1, 2, \ldots K$:

$$\hat{r}_{mgnk} \propto \exp\Big(\mathbb{E}_q[\log \pi_{0k}] + \frac{1}{2}\big(\mathbb{E}_q[\log|\Lambda_k|] + \log|\hat{\phi}^v_{mgnk}| + F_{mgn}^T\hat{\phi}^v_{mgnk}F_{mgn}\big)\Big), \tag{2.43}$$

in which $F_{mgn} \triangleq \frac{1}{\delta^2}C_{mgn}^T(P_{mgn}\hat{x}_m - \hat{u}_{mgn})$. The entire vector $\hat{r}_{mgn}$ is then normalized to sum to one. Each entry k defines the posterior probability (or *responsibility*) that cluster k explains this patch. The required expectations have known closed-form due to our exponential family assumptions. We provide closed-form expressions for $E_q[\pi_{0k}]$ and $E_q[\log|\Lambda_k|]$ in Equation (2.36) and Equation (2.35).

Updating $q(v \mid w, z)$. We update the approximate posterior over the vector $v_{mgn} \in \mathbb{R}^G$ by computing its mean and covariance via closed-form updates:

$$\hat{v}_{mgnk} = \frac{1}{\delta^2}\hat{\phi}^v_{mgnk} C^T_{mgn}(P_{mgn}\hat{x}_m - \hat{u}_{mgn}), \quad \hat{\phi}^v_{mgnk} = \Big(\frac{1}{\delta^2} C^T_{mgn} C_{mgn} + \mathbb{E}_q[\Lambda_k]\Big)^{-1}. \tag{2.44}$$

A closed-form expression for $\mathbb{E}_q[\Lambda_k]$ is given in Equation (2.35). For most patches that are fully-observed, matrix C_{mgn} would just reduce to an identity matrix and the updates simplify accordingly.

Updating $q(u \mid w)$. Similarly, the update for the mean and variance of the scalar offset u_{mgn} is:

$$\hat{u}_{mgn} = \hat{\phi}^u_{mgn}\Big(\frac{r}{s^2} + \frac{1}{\delta^2}\mathbf{1}^T\big(P_{mgn}\hat{x}_m - C_{mgn}\mathbb{E}_q[v_{mgn}]\big)\Big), \quad \hat{\phi}^u_{mgn} = 1\Big/\Big(\frac{1}{s^2} + \frac{D_{mgn}}{\delta^2}\Big). \tag{2.45}$$

The required expectation $\mathbb{E}_q[v_{mgn}]$ is defined in Equation (2.40). $D_{mgn} \in (0, G]$ is the number of observable pixels of patch n in the g-th grid of image m.

2.A.3 Approximate Posterior for Image Random Variable

As the posterior $q(w_m)$ for alignment indicator w_m is assumed uniform, we only need to focus on the approximate posterior $q(x_m)$ for the clean image x_m. In Equation (2.21), we have set $q(x_m)$ to be a Gaussian distribution with mean value $\hat{x}_m$ and covariance matrix $\hat{\phi}^x_m$. As presented in Equation (2.23), this mean and covariance of approximate posterior for the whole-image vector x_m both have closed-form updates. In particular, the covariance update conveniently yields a diagonal matrix.

2.B HDP Grid: Variational Inference Details

While the DP Grid model above assumes the same cluster probability vector π_0 for each image m, our HDP Grid model allows image-specific cluster probabilities π_m to be learned from data. These are tied together via the hierarchical Dirichlet process prior.

2.B.1 Approximate Posterior for HDP Random Variables

Our revised approximate posterior family $\mathcal{Q}$ now includes the HDP factors:

$$q(\beta) = \prod_{k=1}^{\infty} \text{Beta}\left(\beta_k | \hat{\rho}_k \hat{\omega}_k, (1-\hat{\rho}_k)\hat{\omega}_k\right),$$
$$q([\pi_{m1} \dots \ \pi_{mK} \ \pi_{m>K}]) = \text{Dirichlet}(\hat{\theta}_{m1}, \dots, \hat{\theta}_{mK}, \hat{\theta}_{m>K}). \tag{2.46}$$

Here, the image-specific free parameter $\hat{\theta}_m$ is a vector of length $K+1$, where the last dimension represents all inactive clusters. Its optimal update is:

$$\hat{\theta}_{mk} = \begin{cases} \alpha \mathbb{E}_q[\pi_{0k}] + \frac{1}{G} \sum_{g=1}^{G} \sum_{n=1}^{N_{mg}} \hat{r}_{mgnk}, & k \leq K; \\ \\ \alpha \mathbb{E}_q[\pi_{0>K}], & k = K+1. \end{cases} \tag{2.47}$$

$\mathbb{E}_q[\pi_{0k}]$ follows from Equation (2.36) and $\mathbb{E}_q[\pi_{0>K}]$ from Equation (2.37). The update for $\hat{\rho}_k$ and $\hat{\omega}_k$ has no closed form but can be executed easily via gradient descent. Details can be found in Appendix D of the supplement of Hughes et al. [2015], which is available online.*

Other factors remain unchanged from the DP Grid model. Their respective updates remain unchanged as well, except that we substitute $\mathbb{E}_q[\log \pi_{mk}]$ for $\mathbb{E}_q[\log \pi_{0k}]$ in the patch-cluster

*http://michaelchughes.com/papers/HughesKimSudderth_AISTATS_2015_supplement.pdf

responsibility update in Equation (2.43):

$$\hat{r}_{mgnk} \propto \exp\Big(\mathbb{E}_q[\log \pi_{mk}] + \frac{1}{2}\big(\mathbb{E}_q[\log|\Lambda_k|] + \log|\hat{\phi}^v_{mgnk}| + F^T_{mgn}\hat{\phi}^v_{mgnk}F_{mgn}\big)\Big). \tag{2.48}$$

Sparse responsibilities. In practice, we optimize downstream computations by enforcing $\hat{r}_{mgn}$ to be a one-hot vector rather than a dense vector of K entries. To do this, after computing the dense $\hat{r}_{mgn}$ vector as before, we place probability mass one on its maximum entry k'. The advantage of restricting to sparse $\hat{r}$ vectors is that we need only compute and store $\hat{v}_{mgnk'}$ rather than all $k \in \{1, ..., K\}$. Using sparse posteriors significantly reduces memory and computational costs but does not noticeably impact inference quality.

2.B.2 HDP Denoising Algorithm

In Algorithm 2.1, we describe the procedure used to perform our HDP denoising algorithm, which combines K' novel clusters from the noisy test image with the original K clusters learned from a training dataset of clean images. The annealing schedule used to decay δ over 8 iterations from the initial value of the noise-level σ to a final value of $0.5/255$ is equivalent to the schedule used in the public EPLL code.

Algorithm 2.1 HDP denoising algorithm given pre-trained external model

Input:

y_m : noisy image
σ : standard deviation of noise
K' : number of internal clusters to learn from provided image

Output:

$\hat{x}_m$: restored image

function DENOISEIMAGE(y_m)
    Extend $q(\beta)$ and $q(\Lambda)$ to contain $K + K'$ clusters
    Initialize $\mathbb{E}_q[\pi_m]$ as uniform, $\mathbb{E}_q[x_m]$ as y_m, and $\mathbb{E}_q[u_m]$ as the means of y_m patches
    for iteration $t := 1 \to 8$ **do**
        if $t = 1$ **then**
            $\delta := \sigma$
        else
            $\delta := \max\left\{\frac{\sigma}{2^{t/2}}, \frac{0.5}{255}\right\}$
        end if
        for grid $g := 1 \to G$ **do**
            for patch $n := 1 \to N_{mg}$ **do**
                Update $q(z_{mgn})$ using Equation (2.48)
                Update $q(v_{mgn})$ using Equation (2.44)
                Update $q(u_{mgn})$ using Equation (2.45)
            end for
        end for
        Update $q(x_m)$ using Equation (2.23)
        Update $q(\pi_m)$ using Equation (2.47)
        Delete unused image-specific clusters
    end for
    return $\hat{x}_m$
end function

Chapter 3

Stochastic VI for Large-scale Noisy-OR Topic Graphs

In this chapter, we propose a stochastic variational inference algorithm [Hoffman et al., 2013] for training large-scale Bayesian networks, where noisy-OR conditional distributions [Horvitz et al., 1988] are used to capture higher-order relationships. One application is to the learning of hierarchical topic models for text data [Liu et al., 2016]. While previous work has focused on two-layer networks popular in applications like medical diagnosis [Shwe et al., 1991, Jaakkola and Jordan, 1999], we develop scalable algorithms for deep networks that capture a multi-level hierarchy of interactions.

Our key innovation is a family of constrained variational bounds that only explicitly optimize posterior probabilities for the sub-graph of topics most related to the sparse observations in a given document. These constrained bounds have comparable accuracy but dramatically reduced computational cost. Using stochastic gradient updates based on our variational bounds, we learn noisy-OR Bayesian networks orders of magnitude faster than was possible with prior Monte Carlo learning algorithms, and provide a new tool for understanding large-scale binary data. This chapter was previously published as [Ji et al., 2019].

3.1 Introduction

Probabilistic graphical models provide an elegant, interpretable framework for characterizing uncertainty in relationships within high-dimensional data [Koller and Friedman, 2009]. For binary directed graphical models, or Bayesian networks, noisy-OR conditional distributions effectively capture higher-order dependencies for applications including medical diagnosis [Shwe et al., 1991], dimensionality reduction [Šingliar and Hauskrecht, 2006], and text mining [Liu et al., 2016]. Noisy-OR conditionals assume the activity of each variable is independently influenced by each parent, allowing correlations to be modeled with cost linear (rather than exponential) in the degree of each variable node.

While the restricted noisy-OR parameterization improves the efficiency of individual inference algorithm updates, standard methods struggle with web-scale data, where graphs with thousands of variables may be used to model corpora with millions of observations. In this chapter, we develop a rigorous stochastic variational inference algorithm that allows training of noisy-OR Bayesian networks whose scale is orders of magnitude larger. Our approach involves three complementary technical innovations that enable learning of deep graph structures, with many thousands of variable nodes, from very large training databases.

Our first innovation is to develop a family of variational bounds [Wainwright and Jordan, 2008] that is applicable to deep hierarchies of variable relationships. Many prior noisy-OR Bayesian networks, like the classic QMR-DT network for medical diagnosis [Shwe et al., 1991], have a bipartite structure where all hidden (unobserved) variable nodes have no parents. There is an extensive literature on inference and learning algorithms tailored to this limited model family, including but not limited to Jaakkola and Jordan [1999], Šingliar and Hauskrecht [2006], Gogate and Domingos [2010], Halpern and Sontag [2013]. However, such two-layer network structures are obviously limited by the assumption that the hidden "causal" variables are mutually independent. We generalize prior variational bounds for

bipartite noisy-OR networks to support arbitrary directed acyclic graphs, and thus capture hierarchical dependencies among latent topics or causes. Unlike loopy belief propagation, which may be unstable for noisy-OR networks with sparse data [Murphy et al., 1999], our variational updates are always convergent.

Our second innovation enables scalability to graphs with large numbers of variables. Most prior work has focused on models with only hundreds of latent variables, due to limitations in computational speed and memory usage. We show that a rigorous family of constrained variational bounds may be constructed via a "local model" that only explicitly includes topic nodes connected to the set of active (positive) evidence nodes. Regardless of the overall model size, our variational bound may be optimized with cost proportional to the number of active observations; for real-world applications where observations are typically sparse, the computational savings are dramatic.

Our third innovation enables scalability to big training databases. Standard variants of the *expectation maximization* (EM) algorithm, including Monte Carlo EM algorithms [Liu et al., 2016], must process all training data to compute the expected statistics required for each maximization step. For large corpora, each iteration may then take hours or days of computational effort. Moreover, some parameter update schemes require storage of intermediate variables that scales linearly with the number of nodes and training samples [Šingliar and Hauskrecht, 2006], which may lead to very high memory usage. We instead develop a variant of the *stochastic variational inference* [Hoffman et al., 2013] algorithm that incrementally samples small batches of data from the training corpus, uses variational inference to analyze that data given the current model, and then takes a (stochastic) gradient step to improve the weight parameters defining the noisy-OR network. This approach dramatically reduces memory usage and speeds convergence, and because our local models define rigorous variational bounds, the overall stochastic variational inference scheme is guaranteed to converge. We validate our approach using datasets of scientific abstracts from DBLP [Tang et al., 2008]

and restaurant reviews from Yelp, and learn effective models for hundreds of thousands of documents and topics.

3.2 Related Work

The QMR-DT network proposed by Shwe et al. [1991] is a two-layer, bipartite graph created by domain experts capturing how about 600 major diseases influence about 4000 possible symptoms. Each disease has an independent probability of producing each symptom, as integrated via noisy-OR conditionals [Horvitz et al., 1988].

Given an observed set of symptoms, the QMR-DT model is used to infer the posterior probability of each disease. Because exact inference is computationally infeasible, Shwe et al. [1991] used the bipartite network structure to develop a stochastic simulation algorithm. Other Monte Carlo methods like [Gogate and Domingos, 2010] support more general network structures, but become slow for graphs with hundreds of nodes. Alternatively, Jaakkola and Jordan [1999] derive variational upper and lower bounds for the QMR-DT posterior marginals, which we generalize in this work.

Two-layer noisy-OR belief networks (like QMR-DT) are sometimes called BN2O models [Henrion, 1991]. To learn BN2O model parameters from observed data, Šingliar and Hauskrecht [2006] propose a variational EM approach based on the bounds of Jaakkola and Jordan [1999]. Halpern and Sontag [2013] propose an alternative learning algorithm based on the method of moments which avoids local optima of the data log-likelihood, but requires the network to be sufficiently sparse.

It is attractive to generalize BN2O graph structures to deeper hierarchies capturing rich dependencies among hidden topics. Jaakkola and Jordan [2000] consider an alternative family of binary Bayesian networks with conditionals based on logistic regression. Murphy

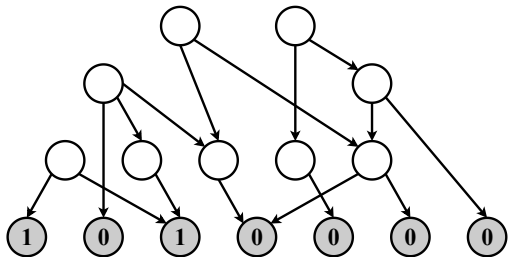


Figure 3.1: Graphical representation of a hierarchical noisy-OR Bayesian network with binary variables. Shaded nodes are observed vocabulary tokens, and their ancestors correspond to hidden topics. The leak node is not shown.

[2012, Section 26.5.4] briefly sketches a deep noisy-OR network used within Google to model the semantic content of text data, but provides few technical details. In this chapter, we generalize the variational bounds of Jaakkola and Jordan [1999] to support multi-layer noisy-OR networks, and formulate extensions that enable learning of large topic graphs from big document corpora.

Liu et al. [2016] also aim to learn general noisy-OR Bayesian networks, but instead propose a Monte Carlo method inspired by the independent cascade model [Wang et al., 2012]. Some aspects of their approach are heuristic: log-likelihoods are scaled by token counts in a way that is not consistent with an underlying generative model, and no theory supports their restriction of sampling updates to document-specific subsets of the topic graph. We include comparisons to variants of their Monte Carlo inference algorithm in Section 3.6.

3.3 Noisy-OR Bayesian Networks

We use binary Bayesian networks as in Figure 3.1 to model vectors of binary features. For the text analysis applications that our experiments focus on, observations are indicators of whether particular tokens (words or phrases) appear in documents. Leaf nodes $j \in \mathcal{O}$ of the network correspond to the vocabulary, where $x_j = 1$ if term j appears in some document. The hidden topic nodes $i \in \mathcal{H}$ have binary variables $z_i \in \{0, 1\}$ indicating whether topics

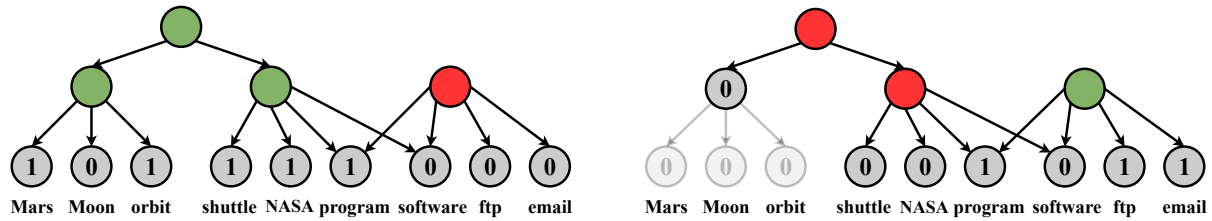


Figure 3.2: Local models for input queries about space science (left) and computer science (right). Our inference algorithm correctly infers the two different meanings of token "program" from its context. Topic nodes with variational probabilities greater than 0.5 are shaded green, and otherwise are shaded red. Some less relevant parts of the local models are not plotted to improve clarity.

appear in that document. For notational simplicity we define a *leak node*, with index 0, that is always active ($z_0 = 1$). It allows some probability of token and topic activation even when other parent nodes are inactive.

Topic nodes are linked by an arbitrary directed acyclic graph, where $\mathcal{P}(i)$ are the parents of node i (excluding the leak node). Hierarchical relationships between topics are captured by the graph structure. Topic activation probabilities are defined by a noisy-OR distribution:

$$p(z_i = 1 \mid z_{\mathcal{P}(i)}) = 1 - \exp\left(-w_{0\to i} - \sum_{k\in\mathcal{P}(i)} w_{k\to i} \cdot z_k\right). \tag{3.1}$$

Activation probabilities for tokens x_j are defined similarly. From Equation (3.1) it follows that the influence of parent nodes factorizes. If parent k is active ($z_k = 1$), it activates child node i with probability $1 - \exp(-w_{k\to i})$, regardless of the states of other parents. If $z_k = 0$, parent k has no influence on the state of z_i. If all parents are inactive, the activation probability $1 - \exp(-w_{0\to i})$ is determined by the leak node.

The noisy-OR structure is useful for reasoning about cases where observations may have multiple hidden causes [Russell and Norvig, 2003]: if a variable is active, then it is likely that at least one of its parents is also active. For example in medical diagnosis, it captures the fact the observed symptoms may be caused by multiple diseases. In hierarchical topic

models it effectively captures polysemy, where a word or phrase may have multiple possible meanings. We provide an example in Figure 3.2.

3.4 Noisy-OR Stochastic Variational Inference

For each document d, we define a variational distribution $q(z^d)$ that factorizes over the hidden topics:

$$q(z^d) \triangleq \prod_{i \in \mathcal{H}} q(z_i^d) = \prod_{i \in \mathcal{H}} (q_i^d)^{z_i^d} (1 - q_i^d)^{1 - z_i^d}. \tag{3.2}$$

Here q_i^d approximates the posterior probability that topic i is active in document d. As the leak node is always on, we fix $q_0^d = 1$. For any $q(z^d)$, the marginal log-likelihood of the observed tokens x^d can be lower bounded by Jensen's inequality as follows:

$$\begin{aligned}
\log p(x^d) &\geq \mathbb{E}_{q(z^d)} \big[\log p(z^d, x^d) - \log q(z^d) \big] \\
&= \sum_{i \in \mathcal{H}} \mathbb{E}_{q(z_i^d,\ z_{\mathcal{P}(i)}^d)} \Big[\log p(z_i^d \mid z_{\mathcal{P}(i)}^d) \Big] + \sum_{j \in \mathcal{O}} \mathbb{E}_{q(z_{\mathcal{P}(j)}^d)} \Big[\log p(x_j^d \mid z_{\mathcal{P}(j)}^d) \Big] \\
&\quad - \sum_{i \in \mathcal{H}} \Big[q_i^d \log q_i^d + (1 - q_i^d) \log (1 - q_i^d) \Big].
\end{aligned} \tag{3.3}$$

Using Equation (3.1), the expectation of the noisy-OR log-probability for each topic can be decomposed as follows:

$$\begin{aligned}
\mathbb{E}_{q(z_i^d,\ z_{\mathcal{P}(i)}^d)} \Big[\log p(z_i^d \mid z_{\mathcal{P}(i)}^d) \Big] =& q_i^d \cdot \mathbb{E}_{q(z_{\mathcal{P}(i)}^d)} \Big[\log \Big(1 - \exp \Big(-w_{0 \to i} - \sum_{k \in \mathcal{P}(i)} w_{k \to i} z_k^d \Big) \Big) \Big] \\
&+ \Big(1 - q_i^d \Big) \cdot \Big(- w_{0 \to i} - \sum_{k \in \mathcal{P}(i)} w_{k \to i} q_k^d \Big).
\end{aligned} \tag{3.4}$$

Due to the non-conjugate structure of the noisy-OR distribution, the expectation in the right side of Equation (3.4) requires enumerating all joint states of the parent nodes, which has

complexity exponential in the node degree. To simplify, we first define the *concave* function

$$f(a) \triangleq \log\big(1 - \exp(-a)\big). \tag{3.5}$$

Because both $w_{0\to i}$ and $w_{k\to i} z_k^d$ are non-negative, we can use Jensen's inequality to derive a lower bound as in Jaakkola and Jordan [1999]:

$$f\big(w_{0\to i} + \sum_{k\in\mathcal{P}(i)} w_{k\to i} z_k^d\big) \geq f(w_{0\to i}) + \sum_{k\in\mathcal{P}(i)} r_{k\to i}^d z_k^d \Big[f(u_{k\to i}^d) - f(w_{0\to i})\Big]. \tag{3.6}$$

Here we introduce an auxiliary parameter $r_{k\to i}^d$ for each non-leak parent edge, with the constraints

$$r_{k\to i}^d \geq 0, \quad \sum_{k\in\mathcal{P}(i)} r_{k\to i}^d = 1, \tag{3.7}$$

and define $u_{k\to i}^d \triangleq w_{0\to i} + w_{k\to i}/r_{k\to i}^d$. We then define a lower bound with complexity *linear* in the node degree:

$$\mathbb{E}_{q(z^d_{\mathcal{P}(i)})}\Big[f\Big(w_{0\to i} + \sum_{k\in\mathcal{P}(i)} w_{k\to i} z_k^d\Big)\Big] \geq f(w_{0\to i}) + \sum_{k\in\mathcal{P}(i)} r_{k\to i}^d q_k^d \Big[f(u_{k\to i}^d) - f(w_{0\to i})\Big]. \tag{3.8}$$

A similar lower bound can be constructed for token nodes' expectations of $\log p(x_j^d \mid z_{\mathcal{P}(j)}^d)$ in Equation (3.3). The overall variational objective for document d is then

$$\begin{aligned}
\mathcal{L}_d(q^d, r^d, w) \triangleq & \sum_{i \in \mathcal{H}} q_i^d \cdot \Big[f(w_{0\to i}) + \sum_{k \in \mathcal{P}(i)} r_{k\to i}^d q_k^d \big(f(u_{k\to i}^d) - f(w_{0\to i}) \big) \Big] \\
& + (1 - q_i^d) \cdot \Big(- w_{0\to i} - \sum_{k \in \mathcal{P}(i)} w_{k\to i} q_k^d \Big) \\
& + \sum_{j \in \mathcal{O}} x_j^d \cdot \Big[f(w_{0\to j}) + \sum_{k \in \mathcal{P}(j)} r_{k\to j}^d q_k^d \big(f(u_{k\to j}^d) - f(w_{0\to j}) \big) \Big] \\
& + (1 - x_j^d) \cdot \Big(- w_{0\to j} - \sum_{k \in \mathcal{P}(j)} w_{k\to j} q_k^d \Big) \\
& - \sum_{i \in \mathcal{H}} \Big[q_i^d \log q_i^d + (1 - q_i^d) \log(1 - q_i^d) \Big].
\end{aligned} \tag{3.9}$$

3.4.1 Expectation Step

In this section, we derive closed-form update equations for local parameters q^d and r^d of each document. For notational simplicity, we omit the document index d.

Fixed Point Update for Edge Parameters r

For every topic and active token node i, we optimize the auxiliary parameters $r_{k\to i}$ given a fixed variational distribution q. Inactive tokens are excluded because if $x_j = 0$, the fourth line of Equation (3.9) has no dependence on $r_{k\to j}$. We show in Appendix 3.A that this optimization problem is concave and has a unique global maximum. Following Jaakkola and Jordan [1999] we derive a fixed-point algorithm by setting the partial derivative of Equation (3.9) to zero after adding a Lagrange multiplier enforcing the normalization constraint of Equation (3.7):

$$r_{k\to i} \propto q_k r_{k\to i} \times \left[f(u_{k\to i}) - f(w_{0\to i}) - \frac{w_{k\to i}}{r_{k\to i}} \cdot f'(u_{k\to i}) \right]. \tag{3.10}$$

Here $f'(a) = \frac{\exp(-a)}{1-\exp(-a)}$ is the derivative of $f(a)$. Because the updates of r for different nodes are independent, the for-loop in line 15 of Algorithm 3.1 may be easily parallelized. This iterative update monotonically increases $\mathcal{L}_d$ and rapidly converges to the global maximum.

Coordinate Update for Node Parameters q

To update the variational posterior q given fixed auxiliary parameters r, we cannot directly use prior work specialized to two-layer noisy-OR networks [Jaakkola and Jordan, 1999]. Instead we directly optimize q by taking the partial derivative of Equation (3.9) and setting to zero:

$$q_i = \frac{1}{1 + \exp\big(-g(q_{\mathcal{P}(i)}, q_{\mathcal{C}(i)}, x, r, w)\big)}. \tag{3.11}$$

Here $\mathcal{C}(i)$ are the children of node i, and

$$\begin{aligned} g(\cdot) \triangleq f(w_{0\to i}) + w_{0\to i} + \sum_{k\in\mathcal{P}(i)} w_{k\to i} q_k + q_k r_{k\to i}\Big(f(u_{k\to i}) - f(w_{0\to i})\Big) \\ + \sum_{\ell\in\mathcal{C}(i)\cap\mathcal{H}} q_\ell r_{i\to\ell}\Big(f(u_{i\to\ell}) - f(w_{0\to\ell})\Big) - (1-q_\ell) w_{i\to\ell} \\ + \sum_{m\in\mathcal{C}(i)\cap\mathcal{O}} x_m r_{i\to m}\Big(f(u_{i\to m}) - f(w_{0\to m})\Big) - (1-x_m) w_{i\to m}. \end{aligned} \tag{3.12}$$

The logistic function in Equation (3.11) ensures $0 < q_i < 1$. The update for node i depends only on the states of its parents and children, not its full Markov blanket (which includes the children's parents), and is thus simpler than computing the posterior required by a Gibbs sampler.

Initialization of Expectation Parameters

The updates for q and r are coupled by the variational objective of Equation (3.9). Our experiments initialize by setting $r_{k\to i} \propto w_{k\to i}$. This corresponds to the optimal solution whenever the activation probabilities q_k for all parent nodes $k \in \mathcal{P}(i)$ are equal.

To prove this, note that optimizing Equation (3.9) with respect to $r_{k\to i}$ is equivalent to maximizing

$$\sum_{k\in\mathcal{P}(i)} r_{k\to i}\Big[f\Big(w_{0\to i} + \frac{w_{k\to i}}{r_{k\to i}}\Big) - f(w_{0\to i})\Big]. \tag{3.13}$$

Given the non-negativity and normalization constraints in Equation (3.7), we can apply Jensen's inequality in the opposite direction of typical variational derivations:

$$\begin{aligned}\sum_{k\in\mathcal{P}(i)} r_{k\to i}\Big[f\Big(w_{0\to i} + \frac{w_{k\to i}}{r_{k\to i}}\Big) - f(w_{0\to i})\Big] &\le f\Big(\sum_{k\in\mathcal{P}(i)} r_{k\to i}\Big(w_{0\to i} + \frac{w_{k\to i}}{r_{k\to i}}\Big)\Big) - f(w_{0\to i}) \\ &= f\Big(w_{0\to i} + \sum_{k\in\mathcal{P}(i)} w_{k\to i}\Big) - f(w_{0\to i}). \end{aligned} \tag{3.14}$$

The bound in the second line of Equation (3.14) is achieved with equality if and only if $w_{0\to i} + \frac{w_{k\to i}}{r_{k\to i}}$ is constant for all parent nodes, which occurs when $r_{k\to i} \propto w_{k\to i}$.

3.4.2 Noisy-OR Weight Optimization

Given optimized local parameters for all data, previous work by Šingliar and Hauskrecht [2006] directly maximizes a (simplified, BN2O model) likelihood bound by solving a non-linear equation for each edge. This requires explicit storage of the E-step results for all documents, and thus has high computation and storage complexity scaling with the product of the number of nodes and documents. We instead employ stochastic gradient updates of the edge weights w, allowing parameter updates to be frequently interleaved with variational

analyses of small batches of documents. Memory usage is also reduced because the variational posteriors for individual documents need not be explicitly stored.

Gradients for Non-leak Edge Weights

From Equation (3.9), the partial derivative of an edge weight between (non-leak) topic node k and a topic node i is

$$\frac{\partial \mathcal{L}_d}{\partial w_{k\to i}} = q_k^d \left(\frac{q_i^d}{1 - \exp(-u_{k\to i}^d)} - 1 \right). \tag{3.15}$$

Similarly, if node k is linked to a token node j, then

$$\frac{\partial \mathcal{L}_d}{\partial w_{k\to j}} = q_k^d \left(\frac{x_j^d}{1 - \exp(-u_{k\to j}^d)} - 1 \right). \tag{3.16}$$

Gradient for Leak Edge Weights

For an edge between leak node 0 and topic node i,

$$\frac{\partial \mathcal{L}_d}{\partial w_{0\to i}} = q_i^d f'(w_{0\to i}) - (1 - q_i^d) + q_i^d \sum_{k\in\mathcal{P}(i)} q_k^d r_{k\to i}^d \big(f'(u_{k\to i}^d) - f'(w_{0\to i}) \big). \tag{3.17}$$

Similarly, if the leak node is linked to a token node j,

$$\frac{\partial \mathcal{L}_d}{\partial w_{0\to j}} = x_j^d f'(w_{0\to j}) - (1 - x_j^d) + x_j^d \sum_{k\in\mathcal{P}(j)} q_k^d r_{k\to j}^d \big(f'(u_{k\to j}^d) - f'(w_{0\to j}) \big). \tag{3.18}$$

Note that the gradient for non-leak edges depends only on the leak edge weight of the child node, but the gradient for leak edges depends on that child's other parents.

Stochastic Gradient Weight Updates

We use a variant of stochastic variational inference [Hoffman et al., 2013], where a stochastic estimate of the gradient of the variational bound is estimated from a mini-batch of sampled data. Due to the non-conjugate noisy-OR likelihood, we optimize a point estimate of the edge weights rather than a full posterior, as Paisley et al. [2012a] did for logistic-normal distributions. The edge weights $w^{(t)}$ at iteration t are updated as follows:

$$w^{(t+1)} = w^{(t)} + \rho_t A \nabla \mathcal{L}_{\mathcal{D}^{(t)}}(w). \tag{3.19}$$

Here $\mathcal{D}^{(t)}$ is the mini-batch of data at iteration t. This stochastic scheme is guaranteed to converge to a local maximum of $\mathcal{L}$ if the learning rate ρ_t satisfies the conditions of Robbins and Monro [1951] and the preconditioner A is positive definite [Paisley et al., 2012a]. To ensure that all weights $w_{k\to i} > 0$, we use a projected gradient ascent algorithm that replaces any negative weights with a small constant: $w_{k\to i}^{(t+1)} \leftarrow \max(w_{k\to i}^{(t+1)}, \epsilon)$.

Our experiments use a constant learning rate ρ as in Mandt et al. [2017]. While the simplest choice for the preconditioner A is the identity matrix, to accelerate convergence we scale the non-leak edges with a constant $c > 1$ so that their magnitudes are more comparable to the leak edges. Relative to more complicated scalings such as the inverse Hessian [Paisley et al., 2012a] or Fisher information matrix [Hoffman et al., 2013], this simple preconditioner is more computationally efficient, while still rapidly converging to high-likelihood models.

3.5 Variational Model Pruning

The stochastic variational inference algorithm of Section 3.4 still requires inference of all variational parameters for each document in the sampled mini-batch. For models defined by large directed graphs, this can have very high computational demands. We thus develop a

Algorithm 3.1 Noisy-OR Stochastic Variational Inference

Input:

$w^{(t)}$: current edge weights
$\mathcal{D}^{(t)}$: data mini-batch for current iteration
$\{N_E, N_Q, N_R\}$: variational hyperparameters
$\{\rho, c\}$: weight update hyperparameters

Output:

$w^{(t+1)}$: updated edge weights

function NoisyORStochasticVariationalUpdate
    Initialize the gradient $\nabla\mathcal{L}_{\mathcal{D}^{(t)}} := 0$
    # Variational Expectation Step
    for instance $d \in \mathcal{D}^{(t)}$ **do**
        Build local model as in Section 3.5.1
        Initialize r^d as in Section 3.4.1
        for $n_e := 1 \to N_E$ **do**
            # Update node parameters
            for $n_q := 1 \to N_Q$ **do**
                for $i \in \mathcal{H}_d$ **do**
                    Update q_i^d using Equation (3.11)
                end for
            end for
            # Update edge parameters
            for $i \in \{\mathcal{H}_d \cup \mathcal{O}_d^+\}$ **do**
                Update $r_{k\to i}^d$ using Equation (3.10), $k \in \mathcal{P}(i)$; repeat N_R times
            end for
        end for
        # Accumulate gradient information
        Compute $\nabla\mathcal{L}_d$ using Equations (3.15, 3.16, 3.17, 3.18)
        $\nabla\mathcal{L}_{\mathcal{D}^{(t)}}$ += $\nabla\mathcal{L}_d/|\mathcal{D}^{(t)}|$
    end for
    # Stochastic Weight Optimization Step
    Apply the gradient update using Equation (3.19)
    return $w^{(t+1)}$
end function

more efficient algorithm that focuses only on document-specific "local models", that contain a small subset of the nodes and edges of the full model. Computation then becomes *sub-linear* in the overall graph size, instead scaling with the number of *active* observations in each document. We first describe how to construct data-dependent local models, and then link to the variational updates of Section 3.4.

3.5.1 Local Model Construction

Our construction of local models is motivated by the observation that real-world observations are typically *sparse*: only a small subset of token nodes are active for each document [Madsen et al., 2005]. For inactive tokens, the posterior probability of their ancestor topics is typically very small. These parts of the graph have little influence on parameter updates because the absolute values of edge weight gradients, as in Equation (3.15) and (3.16), are proportional to topic activation probabilities q_k.

The goal of our local model construction process is to prune these irrelevant subsets of the graph, while still retaining the nodes that contain information crucial to the subsequent parameter update. Specifically, we construct a document-specific local model (as in Figure 3.3) as follows:

1. Select $\mathcal{O}_d^+$, the set of active tokens for document d.
2. Select $\mathcal{H}_d$, the ancestors of nodes in O_d^+ excluding the leak node. We do explicit variational inference updates only for this subset of topic nodes.
3. Select the direct children of $\mathcal{H}_d$, which are a subset of the other topic nodes and the inactive tokens. Constrain their activation probabilities to zero.
4. Link the leak node to all of the other selected nodes.

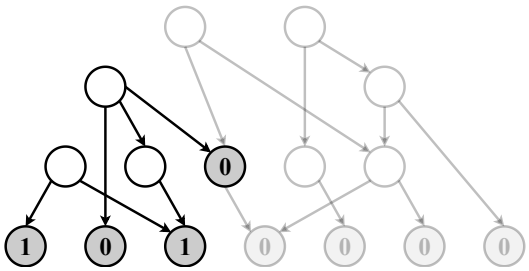


Figure 3.3: By selecting only the nodes most related to a sparse set of active tokens, local models may dramatically reduce the graph size. Here, lightly shaded nodes and edges are pruned. Comparing to the full model of Figure 3.1, explicit inference of activation probabilities is only needed for three of nine topic nodes.

3.5.2 Local Variational Inference

We adapt the stochastic variational inference algorithm of Section 3.4 to the local model defined in Section 3.5.1, dramatically reducing computation and memory demands. Our theoretically sound approach optimizes a constrained family of variational bounds, whose optimum is similar to the original unconstrained variational bound.

Local Model Expectation Step

As can be verified from inspection of Equation (3.9), performing an expectation step with our specified local model is equivalent to constrained variational inference on the full model where we fix $q_i^d = 0$ for all $i \in \mathcal{H} \setminus \mathcal{H}_d$. Adding constraints to the original optimization problem is equivalent to optimizing a lower bound on the original variational objective. As we verify empirically in Section 3.6, because we only apply constraints to topics that have no active descendants, the resulting local bound is a tight approximation. Note that for all $i \in \mathcal{H} \setminus \mathcal{H}_d$, fixing $q_i^d = 0$ also cancels the corresponding auxiliary variables $r_{k \to i}$ in Equation (3.9), which need not be stored or updated.

Local Model Weight Optimization

Although our expectation step is only explicitly performed for local models, we must ensure that gradient updates for all edge weights are still correctly computed. For edges that are included in the local model, we simply use the full model gradient updates from Section 3.4.2.

For non-leak edges outside the local model, each of their parent nodes k must satisfy $k \in \mathcal{H} \setminus \mathcal{H}_d$; if this were not true, then their children would be included in the local model. It thus follows that such parent nodes have activation probability $q_k^d = 0$, and according to Equation (3.15) and (3.16), the resulting gradient will also be exactly zero.

For leak edges outside the local model, the gradient of the edge weights can be shown to equal -1. We verify this by considering two cases. First, if the edge's child node j is a token, it must be inactive ($x_j^d = 0$). All terms scaled by x_j^d in Equation (3.18) then cancel, and only the -1 remains. Alternatively, if the edge's child node i is a topic, then $i \in \mathcal{H} \setminus \mathcal{H}_d$ and $q_i^d = 0$. The partial gradient in Equation (3.17) then simplifies to -1, reducing the prior activation probabilities for topics with no active descendants.

3.6 Experiments

We now evaluate our variational training algorithm on datasets of various scales (see Table 3.1). First, by using a small corpus of newsgroup data where training with the full model is computationally feasible, we illustrate the effectiveness of our local model, the similarity of our variational estimates to expensive Monte Carlo approximations, the influence of hyperparameter c on convergence speed, and qualitative features of learned topic models. Then on two larger datasets, we show that variational training with local models is the only computationally feasible option, and verify the improved efficiency of stochastic variational inference updates.

Table 3.1: Model structure statistics for each dataset

Dataset	# Topics	# Tokens	# Edges
Newsgroups	44	100	707
DBLP	49543	199861	1268551
Yelp	125798	117702	960419

Our learning algorithm assumes the graph structure has already been determined, perhaps via external sources like knowledge bases. As we don't possess such metadata for the text data used in the experiments, we employ a greedy hierarchical clustering method that generalizes the DBScan algorithm [Ester et al., 1996]. It constructs a layered graph structure recursively based on the co-occurrence statistics of token or topic pairs in the previous layer, and also prunes small edges to ensure sparsity. Our approach could be easily integrated with other, more advanced graph learning algorithms.

Unless specified otherwise, we set hyperparameters as follows: $N_E = N_Q = N_R = 10$, $\rho = 0.01$, $c = 1000$.

3.6.1 Tiny 20 Newsgroups

This dataset is a "tiny" version of the famous 20 Newsgroups corpus, with binary occurrence data for 100 words across 16,242 postings.* Each posting (document) is labeled with one of the four highest-level newsgroup categories. Our topic graph contains 44 topic nodes arranged in two layers, as summarized in Table 3.1.

*`https://cs.nyu.edu/~roweis/data/20news_w100.mat`

Table 3.2: Average held-out ELBO and log likelihood of tiny 20 Newsgroups dataset ± two standard deviations

Method	ELBO	Log-Likelihood
VI full	-14.50 ± 0.06	-14.43 ± 0.07
VI local	-14.51 ± 0.08	-14.43 ± 0.07
MCMC full	-15.36 ± 0.15	-14.18 ± 0.07
MCMC local	-19.22 ± 0.47	-17.11 ± 0.12
Initialization	-24.15 ± 0.11	-21.98 ± 0.08

Variational Inference via Local Models

For this small dataset, we compute gradients using the full dataset rather than stochastic mini-batches. 70% of the documents are randomly selected for training. On the remaining 30% we evaluate the average *evidence lower bound* (ELBO), by computing the mean of Equation (3.9) across all test documents; see Table 3.2. The inference algorithm used to evaluate test documents (VI or MCMC, full or local model) is matched to that used during training. The quality of the initialization is assessed using local-model VI. Error bars indicate variability (under the same network structure) across five random train-test splits.

ELBO values in Table 3.2 indicate that our variational inference algorithm, whether using full or local models, increases the log-likelihood bound per document to about -14.5 from the initialization of -24.2. As one verification of the effectiveness of our variational optimization algorithm, these ELBO values are higher than those achieved by MCMC (-15.4), which exactly computes marginal probabilities in the limit where the number of sampling iterations becomes very large [Liu et al., 2016].

More importantly, we find that the difference between the variational bounds achieved by full and local model training is negligible (-14.50 vs -14.51, smaller than the variability from the train-test split). This comparison justifies our use of local models for larger datasets,

where full-model variational inference is prohibitively slow.

As a baseline, we also tried MCMC training using local models constructed in the same way. Compared to using the full model, MCMC test log-likelihoods drop dramatically from -14.2 to -17.1. This deterioration is probably caused by our deterministic procedure for constructing local models, which causes the MCMC edge weight updates to be systematically biased. In contrast, for variational inference local models lead to a principled family of bounds on the overall log-likelihood.

Lower Bounds on Data Log-Likelihood

We also approximately evaluate the marginal log-likelihood of the observed test documents. We construct a simple lower bound by summing up the joint probabilities of all the unique samples drawn over one million iterations of MCMC inference. This lower bound is potentially conservative, because there are $2^{44} \approx 10^{13}$ possible configurations of the latent topic variables. Nevertheless, we verify that our variational objective does bound these approximate log-likelihoods by checking that the MCMC estimates always exceed the corresponding ELBO values in Table 3.2. Previous work demonstrated the accuracy of variational bounds for directed graphical models with discrete hidden variables [Beal and Ghahramani, 2006].

Convergence Speed Acceleration

The preconditioner A was set to an identity matrix when running the preceding experiments. With this choice, thousands of iterations are required for convergence. Empirically, this occurs because the gradients for non-leak edge weights have small magnitude, often about two or three orders of magnitude smaller than the gradients for leak edge weights. To better balance these gradients and improve convergence speed, we explore alternative values for the preconditioning hyperparameter c defined in Section 3.4.2.

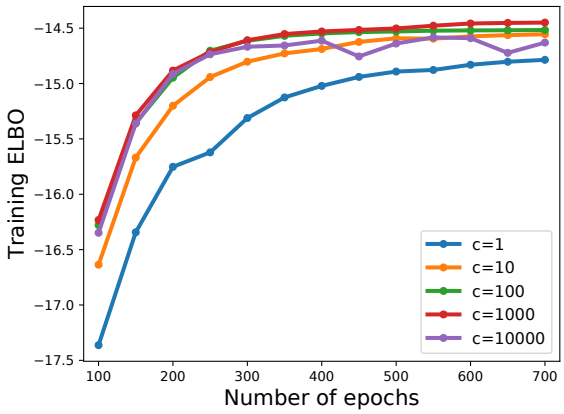


Figure 3.4: Accelerated convergence via hyperparameter c, the scaling of non-leak edge weights in the preconditioner A for stochastic gradient updates (see Section 3.4.2).

As shown in Figure 3.4, the learning algorithm does not converge after hundreds of epochs when $c = 1$. As we increase its value, the magnitudes of leak and non-leak gradients become better balanced, so that convergence becomes much more rapid. The fastest convergence speed is reached when $100 \leq c \leq 1000$. For larger values of c, the step size for non-leak edges becomes too large and optimization may become unstable.

Qualitative and Quantitative Analysis

Qualitatively, running inference on our trained model naturally visualizes the activated topics of input queries. Figure 3.2 shows two examples where the activated topics are each related to space and computer science. In particular, as the token "program" has different meanings for each area, different topics are activated based on the context provided by other tokens. Other tokens like "software" have only one meaning, but may nevertheless be shared among multiple topics. The strength of each relationship in the topic graph is determined by the learned edge weights.

Topic models are sometimes used to define features for document classification and retrieval [Yi and Allan, 2009]. We use the activation probabilities q^d of each document d as a feature representation for classification tasks. One-vs-all linear SVMs [Fan et al., 2008]

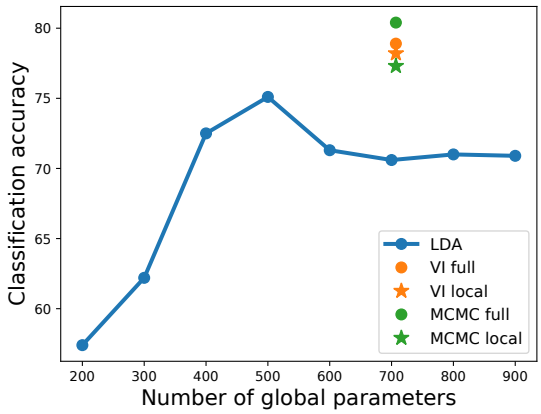


Figure 3.5: Classification accuracy on the tiny 20 Newsgroups dataset for variants of our training algorithm, and Bernoulli LDA models with $2 \leq K \leq 9$ topics.

are trained based on the four newsgroup labels, where the regularization parameter is selected via five-fold cross-validation. To make features more consistent across documents, activation probabilities are standarized within each document by subtracting the mean and dividing by the standard deviation. The baseline model we compare with is *latent Dirichlet allocation* (LDA, Blei et al. [2003]), where multinomial topics are replaced by Bernoulli distributions to model binary observations. For variational training, the numbers of global parameters in LDA is the product of the vocabulary size (100 in this case) and topic count K. Figure 3.5 provides the results when $2 \leq K \leq 9$, which is of similar size to our model that contains 707 edges. The LDA models reach the best performance in this range when K is 4 or 5, corresponding roughly to the 4 newsgroup labels. The different variants of our graph-based learning algorithms are all superior.

3.6.2 DBLP Papers and Yelp Reviews

Now we evaluate our algorithm on two larger datasets. The first one comes from the DBLP bibliography of major computer science publications [Tang et al., 2008].[†] We get 430,213 training documents by extracting paper titles and abstracts in venues for database, data

[†] `http://aminer.org/billboard/aminernetwork`

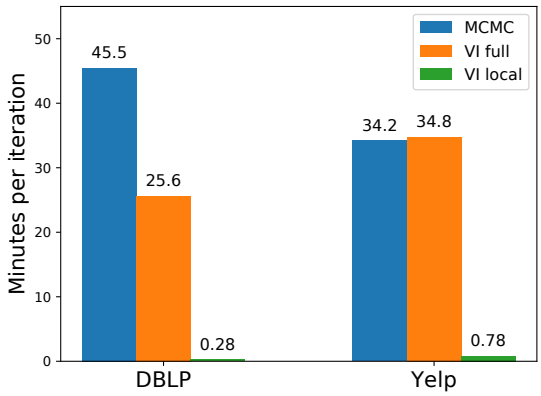


Figure 3.6: Time used for running one E-step iteration on the full-batch data of DBLP and Yelp using 50 CPUs. Local-model VI is the only feasible option for both datasets to run multiple inference iterations till convergence. As restaurant reviews are usually longer than paper titles and abstracts, local models for Yelp tend to be larger than DBLP, and thus need more time for inference.

mining, machine learning, natural language processing, and computer vision research. The other dataset is constructed from the Yelp Open Dataset[‡], where we extract reviews from the top 250 businesses in the "Restaurants" category to produce 483,448 training documents. The tokens for each document are segmented using the method of Liu et al. [2015], which removes both rare and common (stop) words, and also groups words into common phrases. We build a four-layer topic graph for each dataset, whose statistics are summarized in Table 3.1.

For models of these scales, the only computationally feasible option is variational training on local models (Figure 3.6). In Figure 3.7, we compare the convergence speed of full-batch and stochastic training, with mini batches of 50,000 documents. Test documents are the same as in Liu et al. [2016], with 500 paper abstracts for DBLP and 1000 restaurant reviews for Yelp. Each point in the plot represents the average held-out ELBO evaluated using the full model. By interleaving local and global updates more frequently, stochastic training converges much faster than full-batch inference for both datasets.

As gradient-based weight updates are very fast, the variational inference updates in the

[‡]`https://www.yelp.com/dataset`

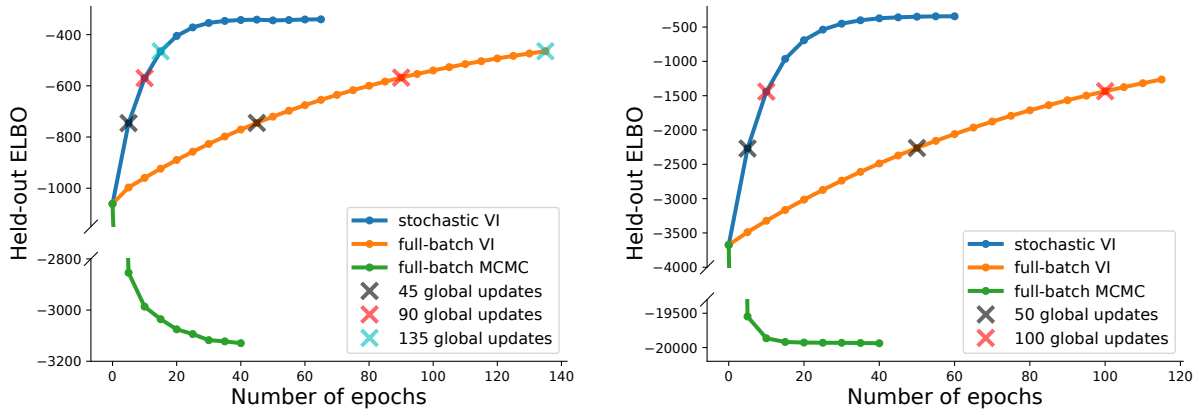


Figure 3.7: ELBO evaluations on the test sets of DBLP (left) and Yelp (right). In both cases, stochastic variational inference converges much faster than standard, full-batch inference. Each pair of `X` markers in the plots compares equal numbers of edge weight updates. Their tiny differences in ELBO values indicates that the noise in stochastic gradient updates does not have a significant impact on the convergence speed. Held-out ELBO values decrease over time for MCMC, likely due to biases caused by its heuristic use of local models. (A regularizer is added to MCMC to avoid edge weights decaying to zero; without this, its performance deteriorates further.)

expectation step dominate computation time. The overhead required by frequent stochastic weight updates is thus negligible.

3.7 Discussion

We have developed a stochastic variational inference algorithm for training large-scale, hierarchical noisy-OR Bayesian networks. We use these models to capture high-order dependencies within the hidden topics and observed tokens in text data. By exploiting the sparsity of input data, our method creates a rigorous variational bound for each document that significantly prunes the model for fast inference. This principled algorithm scales the learning of noisy-OR networks to data and models that are orders of magnitude larger than prior work focusing on simpler, bipartite graphs.

Our algorithms could potentially be used to model causal interactions within many other

types of data, adapted to other model families like the noisy-AND networks used in educational assessment [Conati et al., 1997], or extended to learn graph structures jointly with their parameters [Drton and Maathuis, 2017].

3.A Proof for Concavity of the Noisy-OR Variational Bound in r

For each node $i \in \{\mathcal{H} \cup \mathcal{O}^+\}$ of some document d, the subset of terms in the variational bound of Equation (3.9) that depend on auxiliary variables r_i can be written as:

$$\mathcal{L}_{di}(r_i) = \sum_{k \in \mathcal{P}(i)} r_{k \to i} q_k \Big[f\left(u_{k \to i}\right) - f(w_{0 \to i}) \Big]. \tag{3.20}$$

The first partial derivative of this variational bound is

$$\frac{\partial \mathcal{L}_{di}}{\partial r_{k \to i}} = q_k \Big(f\left(u_{k \to i}\right) - f(w_{0 \to i}) - \frac{w_{k \to i}}{r_{k \to i}} f'(u_{k \to i}) \Big), \tag{3.21}$$

and its second partial derivatives equal

$$\frac{\partial^2 \mathcal{L}_{di}}{\partial r_{k \to i} \partial r_{\ell \to i}} = 0, \quad \frac{\partial^2 \mathcal{L}_{di}}{\partial r_{k \to i}^2} = q_k \frac{w_{k \to i}^2}{r_{k \to i}^3} f''(u_{k \to i}). \tag{3.22}$$

Here, the function

$$f''(a) = \frac{-\exp(a)}{\big(\exp(a) - 1\big)^2} < 0 \tag{3.23}$$

is the second derivative of $f(a)$. Thus on the convex set of auxiliary parameters defined by Equation (3.7), the (diagonal) Hessian matrix of $\mathcal{L}_{di}$ is negative definite, and $\mathcal{L}_{di}(r_i)$ is a strictly concave function of r_i.

Chapter 4

Monte Carlo VI for Probabilistic Programs with Discrete Variables

In this chapter, we propose a broadly applicable variational inference algorithm for probabilistic models with discrete latent variables, using sampling to approximate expectations required for coordinate ascent updates. Applied to three real-world models that use binary variables to capture dependencies within text and image and network data, our approach converges much faster than REINFORCE-style stochastic gradient algorithms [Paisley et al., 2012b, Wingate and Weber, 2013, Ranganath et al., 2014], and requires fewer Monte Carlo samples. Compared to hand-crafted variational bounds with model-dependent auxiliary variables [Gan et al., 2015, Ji et al., 2019], our approach leads to tighter likelihood bounds and greater robustness to local optima. Our Monte Carlo coordinate ascent algorithm is designed to enable easy integration with probabilistic programming languages for effective, scalable, black-box variational inference.

4.1 Introduction

Variational inference is widely used to estimate the posterior distributions of hidden variables in probabilistic models [Wainwright and Jordan, 2008]. Variational bounds are usually

optimized via *coordinate ascent variational inference* (CAVI, Jordan et al. [1999]) algorithms which iteratively update single (or small blocks of) variational parameters. Although CAVI updates can be effective for simple models composed from conjugate priors [Blei et al., 2003], for many models the expectations required for exact CAVI updates are intractable: they may require complex integrals for continuous variables, or computation scaling exponentially with the number of dependent discrete variables.

Variational algorithms for models with non-conjugate conditionals have been derived via hand-crafted auxiliary variables that induce looser, but more tractable, bounds on the data log-likelihood [Jordan et al., 1999, Winn and Bishop, 2005]. Such bounds typically require complex derivations specialized to the parametric structure of specific distributions [Jaakkola and Jordan, 1999, Gan et al., 2015], and thus do not easily integrate with general-purpose inference systems.

To address these limitations, several authors have explored stochastic gradient algorithms that directly optimize a reparameterized bound involving the log-likelihood gradient or score function [Paisley et al., 2012b, Wingate and Weber, 2013, Ranganath et al., 2014], as in the classic REINFORCE policy gradient algorithm [Williams, 1992]. These *black box variational inference* (BBVI) algorithms generalize the stochastic variational inference method of Hoffman et al. [2013] by removing the restriction that variables have conditionally conjugate distributions.

Due to its simplicity and generality, BBVI has become the "standard" variational inference algorithm for a number of probabilistic programming languages including Edward and TensorFlow Probability [Tran et al., 2016, 2018], WebPPL [Goodman and Stuhlmüller, 2014, Ritchie et al., 2016], and Pyro [Bingham et al., 2019]. Unlike other black-box variational methods [Kucukelbir et al., 2017] that require specific variable reparameterizations [Kingma and Welling, 2014], REINFORCE provides unbiased gradients for all models including the

many practically important models with discrete latent variables. However, REINFORCE gradient estimates may have extremely high variance; an official WebPPL tutorial warns that REINFORCE will produce poor results for the LDA topic model [Blei et al., 2003] due to its discrete assignment variables.* They suggest marginalizing discrete variables to produce an alternative model representation where BBVI is more effective, but this requires model-specific derivations that are not generally tractable. Titsias and Lázaro-Gredilla [2015], Tucker et al. [2017], Liu et al. [2019] have proposed variance reduction methods that partially address this issue. But as we demonstrate in this chapter, even for models of moderate size and using control variates to reduce variance, REINFORCE-based variational inference typically requires a large number of iterations (and Monte Carlo samples) for convergence.

In this chapter we analyze the poor convergence behavior of previous BBVI methods in more detail, and contrast it with a Monte Carlo variant of the classic CAVI algorithm. Our Monte Carlo CAVI updates have strong asymptotic guarantees [Ye et al., 2019] while showing good convergence behavior even when few samples are used; in experiments, BBVI typically requires about one hundred times more computation to infer posteriors of comparable quality.

We demonstrate the potential of Monte Carlo CAVI for black-box inference by applying it to diverse models of text and image and network data. Dramatically, in addition to being easier to derive and implement, Monte Carlo CAVI updates are *superior* to previous hand-crafted variational inference algorithms in predictive accuracy and robustness to initialization.

4.2 Probabilistic Models with Binary Latent Variables

We consider probabilistic models that generate observed data x via discrete latent variables z sampled from some joint distribution $p(z, x) = p(z)p(x \mid z)$. For simplicity, we focus our experiments on three models that use binary latent variables to model dependencies within

*`http://probmods.github.io/ppaml2016/chapters/4-3-variational.html`

high-dimensional data x. But as we will show later, all inference algorithms could be easily generalized to latent variables z taking on a larger number of discrete states.

4.2.1 Model One: Noisy-OR Topic Graphs

The first model we consider is the noisy-OR topic graphs explored in Chapter 3:

$$p(z_i = 1 \mid z_{\mathcal{P}(i)}) = 1 - \exp\Big(-w_{0\to i} - \sum\nolimits_{k\in\mathcal{P}(i)} w_{k\to i} \cdot z_k\Big). \tag{4.1}$$

As a quick review, the noisy-OR conditional distribution [Horvitz et al., 1988] assumes the activation of a binary variable is independently influenced by the state of each parent. If parent k is active ($z_k = 1$), it will activate child i with probability $1-\exp(-w_{k\to i})$, regardless of the states of other parents. Inactive parents ($z_k = 0$) have no influence on z_i, and a small "leak" probability $1-\exp(-w_{0\to i})$ allows nodes to occasionally activate even when all parents are off.

On deep noisy-OR Bayesian networks that model topic interactions within documents, we will compare the auxiliary-variable inference technique developed in Chapter 3 with the general-purpose algorithm we are going to introduce in this chapter.

4.2.2 Model Two: Sigmoid Belief Networks

Sigmoid belief networks [Neal, 1992] are layered binary generative models, in which the activation probability of each node is determined by the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$. Following Gan et al. [2015], the activation $z_{i,j}$ of node j in layer i depends on the states of nodes in the preceding layer z_{i+1}:

$$p(z_{i,j} = 1 \mid z_{i+1}) = \sigma(w_{i,j}^T z_{i+1} + c_j). \tag{4.2}$$

```
 1 import torch
 2 from pyro import plate, sample
 3 from pyro.distributions import Bernoulli
 4
 5 class BN(torch.nn.Module):
 6   def __init__(self, params):
 7     super(BN, self).__init__()
 8     self.b, self.W1, self.c1, self.W2, self.c2 = params
 9     self.D_H2, self.D_H1 = self.W2.shape
10
11   @abstractmethod
12   def squash_fun(self, x):
13     raise NotImplementedError
14
15   def model(self, data):
16     dat_axis = plate('dat_axis', data.shape[0], dim=-2)
17     top_axis = plate('top_axis', self.D_H2, dim=-1)
18     mid_axis = plate('mid_axis', self.D_H1, dim=-1)
19     bot_axis = plate('bot_axis', data.shape[1], dim=-1)
20     with dat_axis, top_axis:
21       z_top = sample('z_top', Bernoulli(probs=self.squash_fun(self.b)))
22       wz_top = torch.matmul(z_top, self.W2) + self.c2
23     with dat_axis, mid_axis:
24       z_bot = sample('z_bot', Bernoulli(probs=self.squash_fun(wz_top)))
25       wz_bot = torch.matmul(z_bot, self.W1) + self.c1
26     with dat_axis, bot_axis:
27       sample('x', Bernoulli(probs=self.squash_fun(wz_bot)), obs=data)
28
29 class NoisyOrBN(BN):
30   def squash_fun(self, x):
31     return torch.ones([]) - torch.exp(-x)
32
33 class SigmoidBN(BN):
34   def squash_fun(self, x):
35     return torch.sigmoid(x)
```

Figure 4.1: Pyro implementation of three-layer Bayesian networks. By defining different squashing functions (line 30 and 34), the noisy-OR topic model and sigmoid belief network are easily created from the abstract base class. Variables within "plates" are conditionally independent to each other.

Here, the possibly sparse weight vector $w_{i,j}$ determines which parents directly influence the activation of z_i. In our experiments, two layers of binary latent variables are used to generate pixel values x at the finest scale.

4.2.3 Model Three: Communities and Networks

We consider a simplified version of the nonparametric relational model of Miller et al. [2009], which is used to discover communities from observed social networks. Each entity i is described by a set of D hidden binary features $z_{id} \sim \text{Bernoulli}(\rho)$. The probability that undirected link x_{ij} between entities i and j is present depends on the number of shared features:

$$p(x_{ij} = 1 \mid z) = \Phi\left(w_0 + \sum_{d=1}^{D} w_d z_{id} z_{jd}\right). \tag{4.3}$$

Here, Φ is the CDF of the standard normal distribution, or probit function. The real-valued weight w_d controls the change in link probability when entities share feature d, and $\Phi(w_0)$ is the (small) probability of link occurrence for entities that share no features.

```
import torch
from pyro import plate, sample
from pyro.distributions import Bernoulli

class LFRM(torch.nn.Module):
    def __init__(self, params):
        super(LFRM, self).__init__()
        self.W, self.W0, self.z_prior = params
        self.D = len(self.W)
        self.squash_fun = torch.distributions.normal.Normal(loc=0, scale=1).cdf

    def model(self, links):
        entity_axis = plate("entity_axis", len(links), dim=-2)
        feature_axis = plate("feature_axis", self.D, dim=-1)
        with entity_axis, feature_axis:
            features = sample("features", Bernoulli(probs=self.z_prior))
        idx = torch.triu(torch.ones_like(links), diagonal=1).nonzero()
        with plate("link_axis"):
            wzz = self.W0 + torch.einsum('id,jd->ij', self.W, features**2)[idx]
            sample("links", Bernoulli(probs=self.squash_fun(wzz)), obs=links[idx])
```

Figure 4.2: Pyro implementation of the latent feature relational model. As the model assumes connections are undirected, only the upper triangular part of the link matrix is used as observations.

4.2.4 Probabilistic Programming Languages

Probabilistic programming languages (PPLs) provide flexible but precise frameworks for defining probabilistic models, and performing inference queries given observed data. Popular recent PPLs include Stan [Kucukelbir et al., 2015], Edward and TensorFlow Probability [Tran et al., 2016, 2018], WebPPL [Ritchie et al., 2016], ZhuSuan [Shi et al., 2017], Pyro [Bingham et al., 2019], and Gen [Cusumano-Towner et al., 2019]. Figure 4.1 shows the power of PPLs by defining noisy-OR topic networks and sigmoid belief networks with compact, integrated Pyro code. Figure 4.2 is another example written in Pyro, which specifies the latent feature relational model of Section 4.2.3.

The grand promise of PPLs is that given a generative model specification, appropriate inference code can be automatically generated, enabling rapid model exploration even for non-expert users. But as we show below, existing VI methods for PPLs are often unreliable, and more effective algorithms are sorely needed.

4.3 Existing Variational Inference Algorithms

Exact posterior inference is intractable for models like those in Section 4.2 due to the combinatorial number of latent feature combinations. Mean field variational inference algorithms seek an approximate posterior $q(z)$ from a tractable family with simpler dependencies by maximizing the *evidence lower bound* (ELBO):

$$\mathcal{L}(q) = \mathbb{E}_q \left[\log p(z, x) - \log q(z)\right] \leq p(x). \tag{4.4}$$

Maximizing $\mathcal{L}(q)$ minimizes the Kullback-Leibler divergence from the true posterior $p(z \mid x)$. Many previous studies have found that VI can have dramatic computational advantages compared to MCMC methods like Gibbs samplers [Gopalan and Blei, 2013, Gan et al., 2015,

Gopalan et al., 2016, Ji et al., 2019].

We make a "naïve" mean-field approximation so that $q(z) = \prod_i q(z_i)$ is fully factorized. The following sections review three classic ways to optimize the ELBO, and discuss advantages and drawbacks that motivate the Monte Carlo VI method of Section 4.4.

4.3.1 Coordinate Ascent Variational Inference

Coordinate ascent variational inference (CAVI) is the standard approach to optimizing mean field variational bounds [Jordan et al., 1999, Winn and Bishop, 2005, Blei et al., 2017]. It iteratively optimizes each factor of the variational density while holding all others fixed, producing iterations that monotonically increase the ELBO and converge to a (local) maximum.

Concretely, to update variational factor $q(z_i)$, CAVI requires the *complete conditional* $p(z_i \mid z_{-i}, x)$ of z_i given all other latent variables z_{-i} and the observations x. The optimal $q(z_i)$ then equals

$$q(z_i) \propto \exp\left\{\mathbb{E}_{-i}[\log p(z_i \mid z_{-i}, x)]\right\}, \tag{4.5}$$

where the expectation is with respect to the current iteration's variational distributions $q(z_{-i}) = \prod_{j \neq i} q(z_j)$.

For binary variables, $q(z_i)$ is a Bernoulli distribution. If we use the logit $\tau_i \triangleq \log \frac{q(z_i=1)}{q(z_i=0)}$ as the variational parameter for $q(z_i)$, then Equation (4.5) simplifies to

$$\tau_i = \mathbb{E}_{-i}\left[\log \frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\right]. \tag{4.6}$$

While CAVI provides a uniform way to optimize the ELBO, it is not computationally tractable for many models with high-degree variable relationships. In particular, for non-

conjugate conditionals like those in Equation (4.1), (4.2), and (4.3), computing the expectations in Equation (4.6) requires enumerating the exponentially many joint configurations of variables in the Markov blanket of z_i.

4.3.2 Auxiliary Variable Inference Methods

Tractable variational updates have been hand-designed for specific models by crafting bounds, parameterized by auxiliary variables, for challenging conditional distributions. The resulting bounds are looser than the ELBO of Equation (4.4), but may lead to simpler, closed-form variational parameter updates.

Noisy-OR Log Concavity

As discussed in Section 3.4, Jaakkola and Jordan [1999] apply Jensen's inequality and derive a lower bound to the ELBO of Equation (4.4) by leveraging the log-concavity of the noisy-OR function. The new bound becomes a linear function of parent states z_k, so that the update for $q(z)$ has a simple closed form. Moreover, we have proved in Appendix 3.A that the optimization with respect to the newly-introduced auxiliary variable is a concave problem, and the the global maximum can be computed via fixed-point iterations.

Pólya-Gamma Data Augmentation

The Pólya-Gamma data augmentation strategy [Polson et al., 2013] exploits a representation of binomial likelihoods, parametrized by log-odds, as Gaussian scale mixtures with respect to a Pólya-Gamma distribution. In particular, if $\gamma \sim \mathrm{PG}(b, 0), b > 0$, then

$$\frac{(e^{\psi})^a}{(1+e^{\psi})^b} = 2^{-b} e^{(a-b/2)\psi} \int_0^{\infty} e^{-\gamma\psi^2/2} \mathrm{PG}(\gamma|b, 0)\ \mathrm{d}\gamma. \tag{4.7}$$

While the derivation of this representation is challenging, Gan et al. [2015] show that setting $b = 1$ leads to a tractable lower bound for the logistic log-likelihoods in sigmoid belief networks:

$$\begin{aligned}\log p(z_{i,j} \mid z_{i+1}) &= \log \sigma(w_{i,j}^T z_{i+1} + c_j) \\ &\geq -\log 2 + (z_{i,j} - 0.5)(w_{ij}^T z_{i+1} + c_j) - 0.5 \cdot \gamma_{i,j} \cdot (w_{ij}^T z_{i+1} + c_j)^2 \\ &\quad + \mathbb{E}_{q(\gamma_{i,j})}[\log \mathrm{PG}(\gamma_{i,j} \mid b, 0) - \log q(\gamma_{i,j})], \end{aligned} \tag{4.8}$$

which only takes linear (rather than exponential) time to compute the expectation with respect to $q(z)$. The optimal variational distribution for each augmented variable $\gamma_{i,j}$ further turns out to have an analytic form:

$$q(\gamma_{i,j}) = \mathrm{PG}\left(1, \sqrt{\mathbb{E}_{q(z_{i+1})}[(w_{ij}^T z_{i+1} + c_j)^2]}\right) \approx \mathrm{PG}\left(1, w_{ij}^T \mathbb{E}_{q(z_{i+1})}[z_{i+1}] + c_j\right). \tag{4.9}$$

Probits from Thresholded Gaussians

Albert and Chib [1993] show that probit regression models for binary outcomes can be represented by thresholding the output of normal, Gaussian regression models:

$$\Phi(t)^s (1 - \Phi(t))^{1-s} = \int 1\{y \geq 0\}^s 1\{y < 0\}^{1-s} \mathcal{N}(y \mid t, 1)\, \mathrm{d}y. \tag{4.10}$$

This is equivalent to introducing an auxiliary latent variable $y \sim \mathcal{N}(t, 1)$ to the original probabilistic model.

For the latent feature relational model discussed in Section 4.2.3, assume N is the number of entities and D is the feature dimension. The mean-field variational distribution for the latent features is $q(z) = \prod_{i=1}^N \prod_{d=1}^D q(z_{id})$. Each dimension z_{id} is a Bernoulli distribution $q(z_{id}) \sim \mathrm{Bernoulli}(q_{id})$, with the activation probability $q_{id} \triangleq q(z_{id} = 1)$ as the free parameter.

According to Equation (4.4), the ELBO without auxiliary variables is

$$
\begin{aligned}
&\mathcal{L}(q(z)) \qquad\qquad (4.11)\\
=&\mathbb{E}_{q(z)}\Bigg[\sum_{i=1}^{N}\sum_{d=1}^{D} z_{id}(\log\rho - \log q_{id}) + (1-z_{id})(\log(1-\rho) - \log(1-q_{id}))\\
&+\sum_{j>i}^{N} x_{ij}\log\Phi\Big(w_0 + \sum_{d=1}^{D} w_d z_{id} z_{jd}\Big) + (1-x_{ij})\log\Big(1-\Phi\Big(w_0+\sum_{d=1}^{D} w_d z_{id} z_{jd}\Big)\Big)\Bigg].
\end{aligned}
$$

By using the data augmentation trick of Equation (4.10), we introduce an auxiliary variable y_{ij} for each pair of entities. Then the second row of Equation (4.11) would be equivalent to

$$
\begin{aligned}
&\sum_{i=1}^{N}\sum_{j>i}^{N} \log\int 1\{y_{ij}\geq 0\}^{x_{ij}} 1\{y_{ij}<0\}^{1-x_{ij}} \mathcal{N}(y_{ij}|w_0+\sum_{d=1}^{D} w_d z_{id} z_{jd}, 1)\,\mathrm{d}y_{ij}\\
\geq&\sum_{i=1}^{N}\sum_{j>i}^{N} \mathbb{E}_{q(y_{ij})}\left[\log\frac{p(x_{ij}|y_{ij})p(y_{ij}|z)}{q(y_{ij})}\right], \qquad (4.12)
\end{aligned}
$$

in which $p(x_{ij}|y_{ij}) \triangleq 1\{y_{ij}\geq 0\}^{x_{ij}} 1\{y_{ij}<0\}^{1-x_{ij}}$, and $p(y_{ij}|z) \triangleq \mathcal{N}(y_{ij}|w_0+\sum_{d=1}^{D} w_d z_{id} z_{jd}, 1)$. The greater-than-or-equal-to sign in Equation (4.12) comes from applying Jensen's inequality to the log function.

Bringing Equation (4.12) back to Equation (4.11), we get a lower bound of the original ELBO. It is also mathematically equivalent to the ELBO of a data-augmented model in which a latent variable y_{ij} is added to each pair of entities. Following Equation (4.5), the optimal coordinate-ascent variational factor of y_{ij} then obeys a truncated normal distribution:

$$
q(y_{ij}) = \begin{cases} \mathcal{TN}_{+}\big(y_{ij} \mid w_0 + \sum_d w_d q_{id} q_{jd}, 1\big), \text{ if } x_{ij}=1; \\ \\ \mathcal{TN}_{-}\big(y_{ij} \mid w_0 + \sum_d w_d q_{id} q_{jd}, 1\big), \text{ if } x_{ij}=0. \end{cases} \qquad (4.13)
$$

From Equation (4.12) we can see the data-augmented ELBO is a quadratic function of z, so

the coordinate update for the latent feature $q(z)$ can be computed efficiently in quadratic time:

$$q_{id} = \Phi\Big(\log \rho - \log(1-\rho) + \sum_{j \neq i} q_{jd} w_d \big(\mathbb{E}_{q(x_{ij})}[x_{ij}] - w_0 - \frac{1}{2} w_d - \sum_{e \neq d} w_e q_{ie} q_{je} \big) \Big), \quad (4.14)$$

in which the mean of a truncated normal $\mathbb{E}_q[x_{ij}]$ is efficiently computed by calling the function for evaluating unit Gaussian CDF provided in software libraries.

4.3.3 REINFORCE Variational Gradients

REINFORCE is a policy gradient method for reinforcement learning [Williams, 1992] that has been adapted for gradient-based variational inference [Paisley et al., 2012b, Wingate and Weber, 2013, Ranganath et al., 2014]. It optimizes Equation (4.4) via stochastic gradient ascent, computing unbiased ELBO gradients via Monte Carlo samples $z^{(m)}$ drawn from $q(z)$. To avoid constraints, for binary latent variables, noisy gradient updates are parameterized via variational logits τ:

$$\nabla_\tau \mathcal{L} \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_\tau \log q\big(z^{(m)}\big) \cdot \Big(\log p(z^{(m)}, x) - \log q(z^{(m)}) \Big). \quad (4.15)$$

REINFORCE has also been called *black box variational inference* (BBVI, Ranganath et al. [2014]) because it can be applied to different probabilistic models without specialized derivations. We use these two terms interchangeably. Unlike some other variational methods [Kucukelbir et al., 2017], BBVI has the advantage that it can be directly applied to discrete variables which are not easily reparameterized [Kingma and Welling, 2014] via Gaussian latent variables.

BBVI is known to produce high-variance gradient estimates, and many variance reduction tricks have been proposed. Rao-Blackwellized estimators analytically marginalize variables

outside the Markov blanket of the variable being updated, provably reducing variance [Ranganath et al., 2014]:

$$\frac{\partial \mathcal{L}}{\partial \tau_i} \approx \frac{1}{M} \sum_{m=1}^{M} \left. \frac{\partial \log q(z_i)}{\partial \tau_i} \right|_{z_i^{(m)}} \cdot \Big(\log p(z_i^{(m)} \mid z_{-i}^{(m)}, x) - \log q(z_i^{(m)}) \Big). \tag{4.16}$$

Here, $p(z_i \mid z_{-i}, x)$ is the complete conditional defined in Section 4.3.1. Letting $q_i \triangleq q(z_i = 1)$, the score function $\frac{\partial \log q(z_i)}{\partial \tau_i}$ may be computed via the chain rule:

$$\frac{\partial \log q(z_i)}{\partial \tau_i} = \frac{\partial \log q(z_i)}{\partial q_i} \frac{\partial q_i}{\partial \tau_i} = (1 - q_i)^{z_i} (-q_i)^{1 - z_i}. \tag{4.17}$$

For all experiments, our BBVI results use Rao-Blackwellized gradient estimates to reduce variance.

Gradient variance may be further reduced by introducing control variates [Paisley et al., 2012b] that preserve target expectations, but approximately cancel noise to reduce variance. Wingate and Weber [2013], Ranganath et al. [2014], and Ritchie et al. [2016] all set the control variate to be the zero-mean score function scaled by a carefully-chosen constant a_i, which is also called the baseline [Greensmith et al., 2004]:

$$\frac{\partial \mathcal{L}}{\partial \tau_i} \approx \frac{1}{M} \sum_{m=1}^{M} \left. \frac{\partial \log q(z_i)}{\partial \tau_i} \right|_{z_i^{(m)}} \cdot \Big(\log p(z_i^{(m)} \mid z_{-i}^{(m)}, x) - \log q(z_i^{(m)}) - a_i \Big). \tag{4.18}$$

We evaluate this control variate in our experiments.

4.4 Monte Carlo CAVI

4.4.1 A Low-variance Black-box VI Framework

We now propose and evaluate a Monte Carlo approximation to the classical CAVI algorithm of Section 4.3.1, which uses sampling to approximate the expectations needed for optimal variational parameter updates. Recent work by Ye et al. [2019] proves that given appropriate regularity conditions, a Monte Carlo CAVI recursion gets arbitrarily close to a maximizer of the ELBO in Equation (4.4) with any target probability. They apply Monte Carlo CAVI to analyze nuclear magnetic resonance spectroscopy data, and design a Metropolis-within-Gibbs stochastic proposal tailored to a specialized family of continuous-variable models.

We instead assess Monte Carlo CAVI as a general purpose inference framework for models with discrete variables. Approximating the expectations in Equation (4.6) by M samples of the variables in the Markov blanket, the Monte Carlo CAVI update for the logit parameters is

$$\begin{aligned}\tau_i &\approx \frac{1}{M}\sum_{m=1}^{M} \log \frac{p\Big(z_i = 1 \mid z_{-i}^{(m)}, x\Big)}{p\Big(z_i = 0 \mid z_{-i}^{(m)}, x\Big)} \\ &= \frac{1}{M}\sum_{m=1}^{M} \log \frac{p\Big(z_i = 1 \mid u_{\mathcal{P}(i)}^{(m)}\Big) \cdot \prod_{j \in \mathcal{C}(i)} p\Big(u_j^{(m)} \mid z_i = 1, u_{\mathcal{P}(j)}^{(m)}\Big)}{p\Big(z_i = 0 \mid u_{\mathcal{P}(i)}^{(m)}\Big) \cdot \prod_{j \in \mathcal{C}(i)} p\Big(u_j^{(m)} \mid z_i = 0, u_{\mathcal{P}(j)}^{(m)}\Big)},\end{aligned} \tag{4.19}$$

where $u^{(m)} \triangleq \{z_{-i}^{(m)} \cup x\}$ and $\mathcal{C}(i)$ is the set of children of variable i. The second line of Equation (4.19) shows more explicitly how the model structure is leveraged to avoid unnecessary computations with variables outside the Markov blanket. Importantly, the complexity of Monte Carlo CAVI is linear in the number of samples even for models with high-order dependencies.

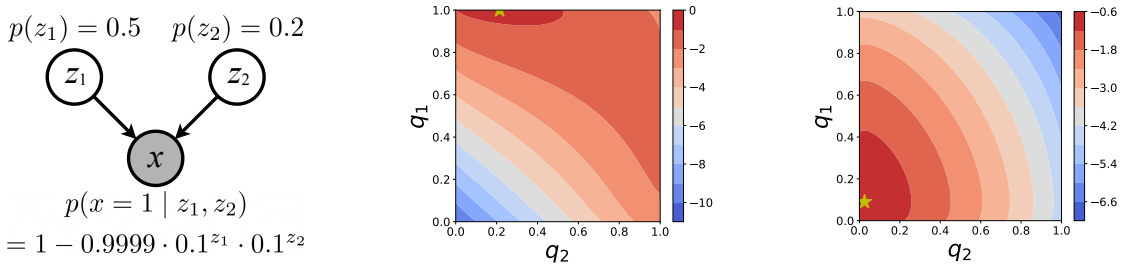


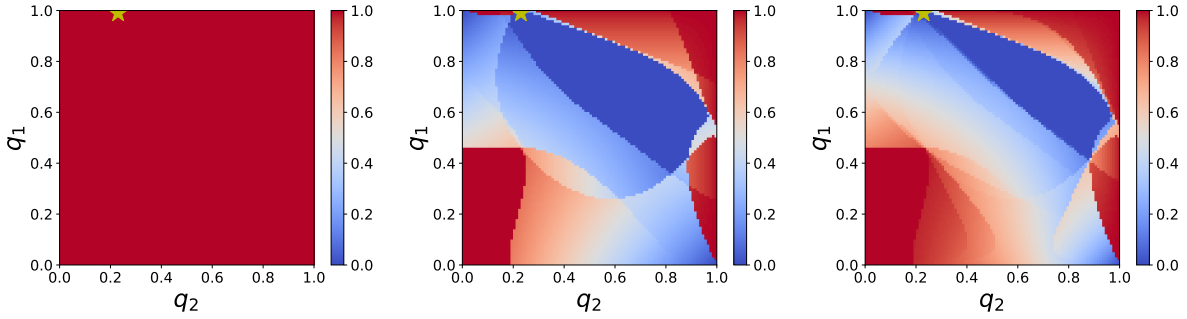
Figure 4.3: A toy noisy-OR model with two latent nodes. *Left:* Graphical representation and probability distributions of nodes in the toy model. *Middle:* Contour plot of the toy model's ELBO as a function of the variational parameters q_1 and q_2, when the observation $x = 1$. The yellow star indicates the global optimum. The likelihood is symmetric with respect to z_1 and z_2, but the prior probability of z_1 is higher, so the optimal $q_1 \approx 0.99$ to explain the active observation $x = 1$. The optimal q_2 is similar to its prior. *Right*: Contour plot when the observation $x = 0$. Variational posteriors q_1 and q_2 are both close to zero in this case. Moreover, the ELBO is a concave function of q.

4.4.2 Comparison with REINFORCE

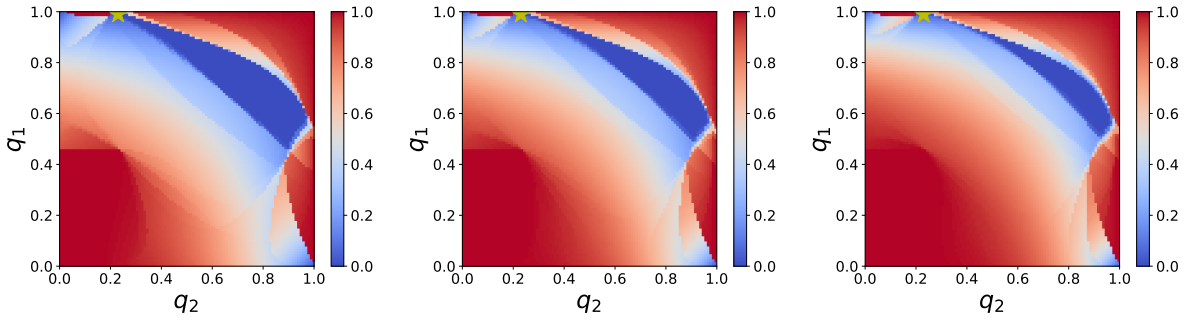
For binary-valued probabilistic models, Monte Carlo CAVI (CAVI for short) and REINFORCE (BBVI for short) are both general-purpose tools for inference. In this section, we show the advantages of CAVI against BBVI using a toy example, as presented in Figure 4.3.

When the observation $x = 1$, this is a simple inference problem without any local optima. Computing the numerical gradient of the logits τ requires enumerating the $2^2 = 4$ possible combinations of z_1 and z_2. As shown in Figure 4.6, among the three different options, BBVI performs best when the learning rate is 1. Under this learning rate, the analytic gradient always increases the ELBO, no matter where the current value of q is, as illustrated in Figure 4.4 (a).

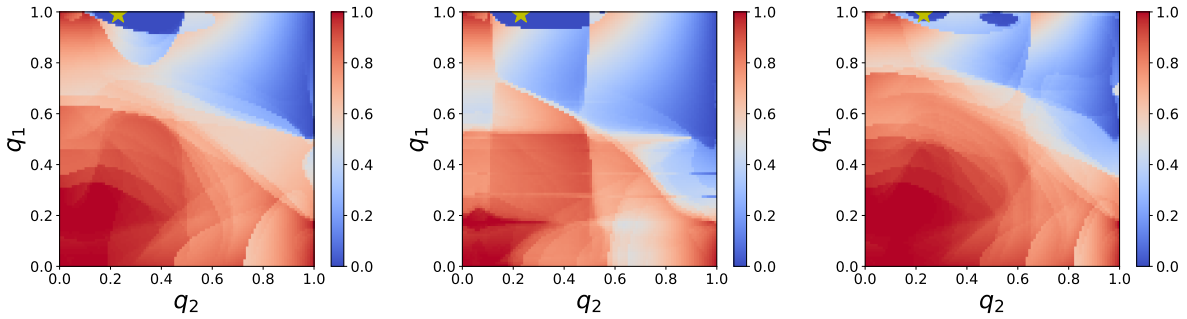
However, because of the randomness caused by Monte Carlo sampling, this is not the case for BBVI, as shown in Figure 4.4(b)-(d). Although the ELBO improvement does get better as the number of samples increases, there is always a "blue belt" around the optimal value, in which the Monte Carlo gradient would change the variational parameters in a wrong direction with over half of the probability. Only with a small probability does the gradient



(a) deterministic gradient (b) BBVI: 1 sample (c) BBVI: 2 samples



(d) BBVI: 3 samples (e) BBVI: 4 samples (f) BBVI: 5 samples

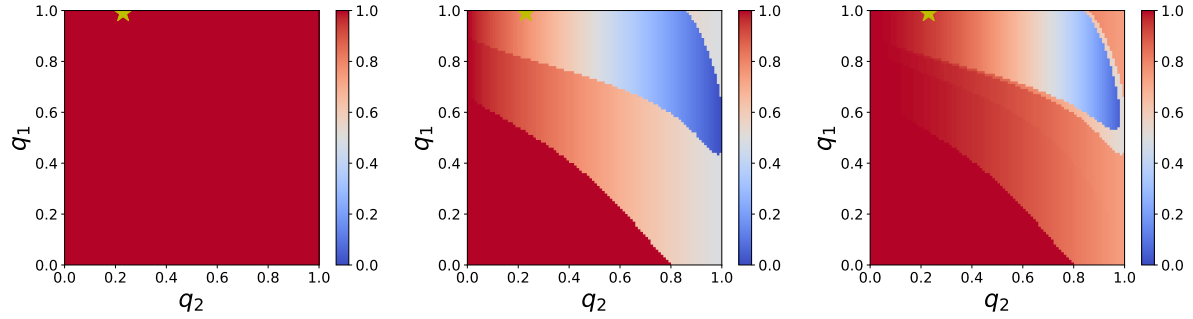


(g) BBVI: 2 sp & 2 cv sp (h) BBVI: 2 sp & 3 cv sp (i) BBVI: 3 sp & 2 cv sp

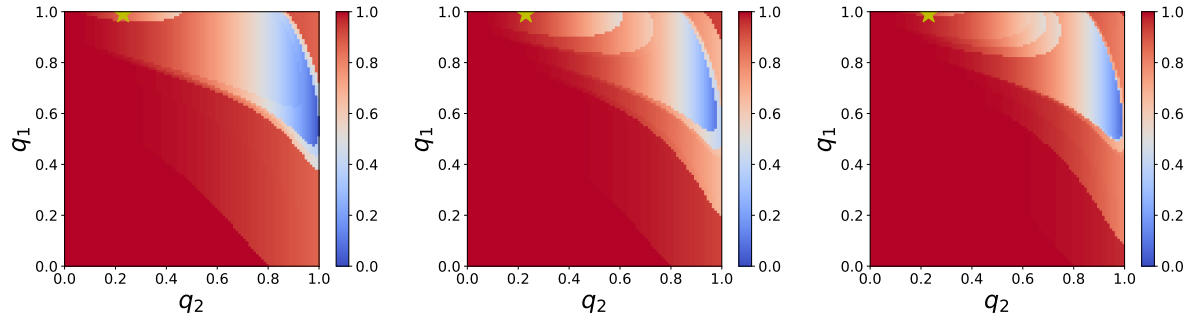
Figure 4.4: The probability of ELBO increase after a gradient update of the logits of q, when the observation $x = 1$. Each point of the plots indicates the current value of q_1 and q_2. (a) the true gradient; (b)-(f) noisy gradients estimated with various numbers of samples; (g)-(i) noisy gradients with extra samples used to estimate the baseline value of control variate (cv). The learning rate is 1.

gets lucky enough to point in a direction that enhances the ELBO (with a large magnitude, so that in expectation BBVI is still unbiased).

Note that adding the control variate is not able to entirely solve this problem. As in Ranganath et al. [2014], we use extra samples to estimate the baseline values a_i in Equa-



(a) regular coordinate update (b) CAVI: 1 sample (c) CAVI: 2 samples



(d) CAVI: 3 samples (e) CAVI: 4 samples (f) CAVI: 5 samples

Figure 4.5: The probability of ELBO increase after a coordinate update of the logits of q, when the observation $x = 1$. (a) the regular coordinate ascent update where expectations are computed numerically; (b)-(f) Monte Carlo CAVI using various numbers of samples.

tion (4.18). The blue area around the global optimum still exists in Figure 4.4(e)-(f).

On the other hand, from Figure 4.5 one can see that Monte Carlo CAVI behaves better than BBVI under the same sampling budgets, in the sense that most areas have higher probability of ELBO improvement. There is also a low-probability region in each plot, but they are much farther away from the global optimum comparing to the ones in BBVI.

More importantly, unlike the gradient-based BBVI method that needs to follow the gradient direction step by step, CAVI updates change the variational distribution of each variable directly to the optimal point, and do not need to tune the step size parameters. As shown in Figure 4.6, even if q is initialized in the blue region unluckily, after only a couple of iterations the CAVI algorithm would escape from it and rapidly converge to the global optimum. On the contrary, because of the high-variance issue around the optimum value, even with a

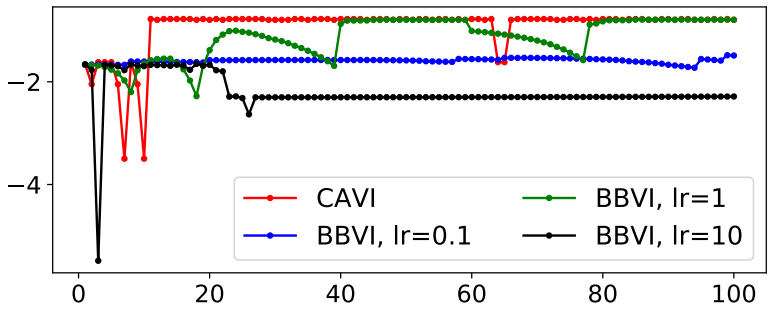


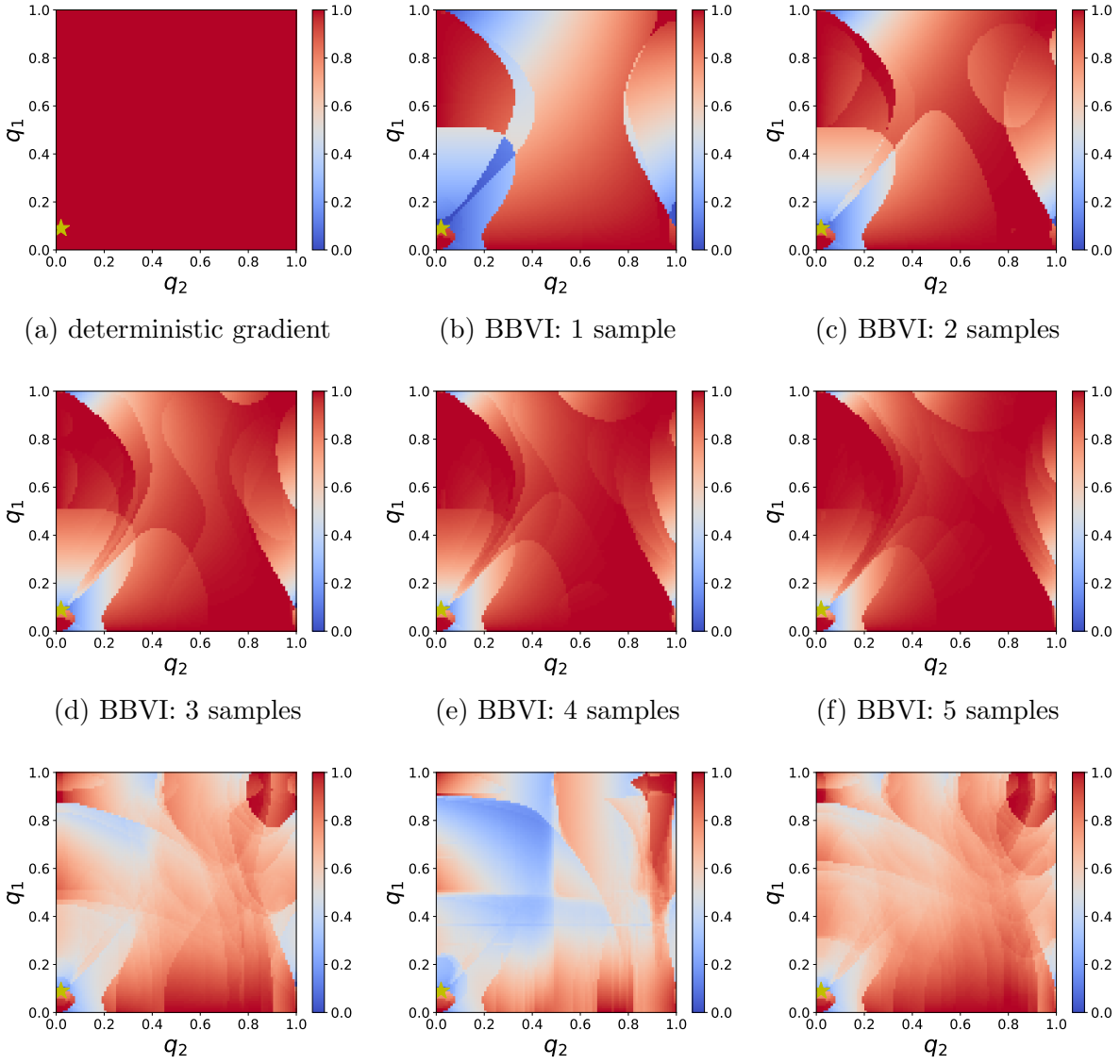
Figure 4.6: ELBO (y axis) over iterations (x axis) on the toy model. The initial value $q_1 = 0.5, q_2 = 0.9$ is selected in the low-probability region in Figure 4.5 intentionally, to see how CAVI works under a bad condition. As expected, its ELBO (red) fluctuates in the first few iterations, but then quickly moves to the optimum. For BBVI, the best learning rate (lr) is 1 in this case. One could see clearly the ubiquitous drops in ELBO across iterations (green). When lr is set too large (black), q would go to extreme areas near the edges or corners. In those areas, the Monte Carlo score function presented in Equation (4.17) is close to zero. Therefore the gradient magnitude becomes so small that it cannot get q back to normal values. The number of samples used for each method is 2.

carefully tuned learning rate, BBVI still needs a great many of iterations to go across the blue region to converge. See Appendix 4.A for more visualizations of this example.

Similar trends are observed on this toy model when we set the observation to be inactive ($x = 0$), as presented in Figure 4.7 and Figure 4.8. In fact, this is a special case where CAVI improves the ELBO with probability one in all areas. That's because the expected log joint $\mathbb{E}_q[\log p(z, x)]$ of the noisy-OR is a linear function of q when $x = 0$. Then the coordinate update equations all become deterministic, requiring no Monte Carlo estimation at all.

4.4.3 Generalization to Non-binary Models

It is straightforward to generalize our Monte Carlo CAVI algorithm to all models with finite-state discrete variables. Assume $\mathcal{V}_i$ is the set of values a latent variable z_i can take, and its size $|\mathcal{V}_i| \geq 2$. As in Blei et al. [2003] and Beal [2003], the normalization constraints of $q(z_i)$ can be enforced by adding Lagrange multiplier terms. For regular CAVI algorithm, the



(a) deterministic gradient (b) BBVI: 1 sample (c) BBVI: 2 samples

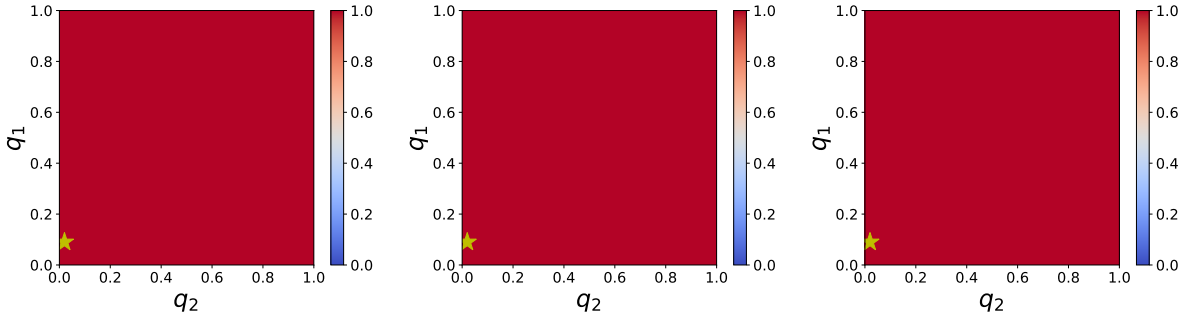
(d) BBVI: 3 samples (e) BBVI: 4 samples (f) BBVI: 5 samples

(g) BBVI: 2 sp & 2 cv sp (h) BBVI: 2 sp & 3 cv sp (i) BBVI: 3 sp & 2 cv sp

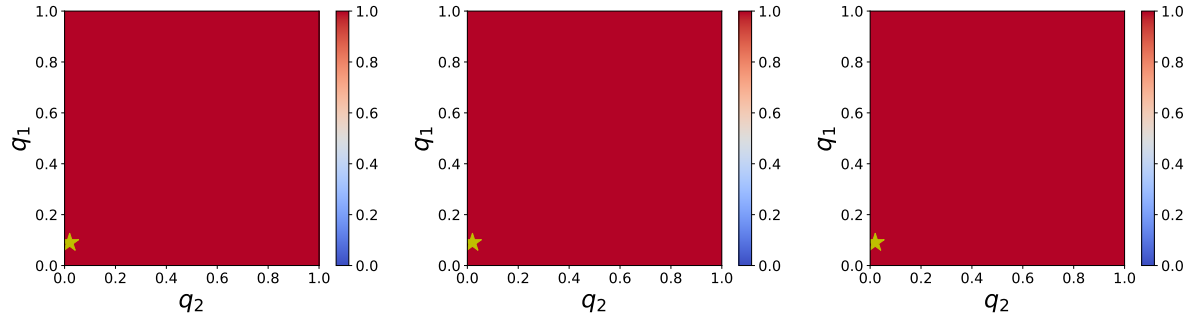
Figure 4.7: The probability of ELBO increase after a gradient update of the logits of q_1 and q_2, when $x = 0$. The learning rate is 1.

variational update for $q(z_i = v)$ where $v \in \mathcal{V}_i$ would be

$$q(z_i = v) \propto \exp\left\{\mathbb{E}_{-i}\big[\log p(z_i = v \mid z_{-i}, x)\big]\right\} \tag{4.20}$$



(a) regular coordinate update (b) CAVI: 1 sample (c) CAVI: 2 samples



(d) CAVI: 3 samples (e) CAVI: 4 samples (f) CAVI: 5 samples

Figure 4.8: When $x = 0$, the expected log joint $\mathbb{E}_q[\log p(z, x)]$ becomes a linear function of q. The coordinate update equations become deterministic and always improve the ELBO with probability one.

When using sampling to estimate the expectation term in Equation (4.20), we get the Monte Carlo CAVI update

$$q(z_i = v) \propto \exp\left\{\frac{1}{M}\sum_{m=1}^{M} \log p\Big(z_i = v \mid z_{-i}^{(m)}, x\Big)\right\} \tag{4.21}$$

$$= \exp\left\{\frac{1}{M}\sum_{m=1}^{M} \log p\Big(z_i = v \mid u_{\mathcal{P}(i)}^{(m)}\Big) + \sum_{j \in \mathcal{C}(i)} \log p\Big(u_j^{(m)} \mid z_i = v, u_{\mathcal{P}(j)}^{(m)}\Big)\right\}$$

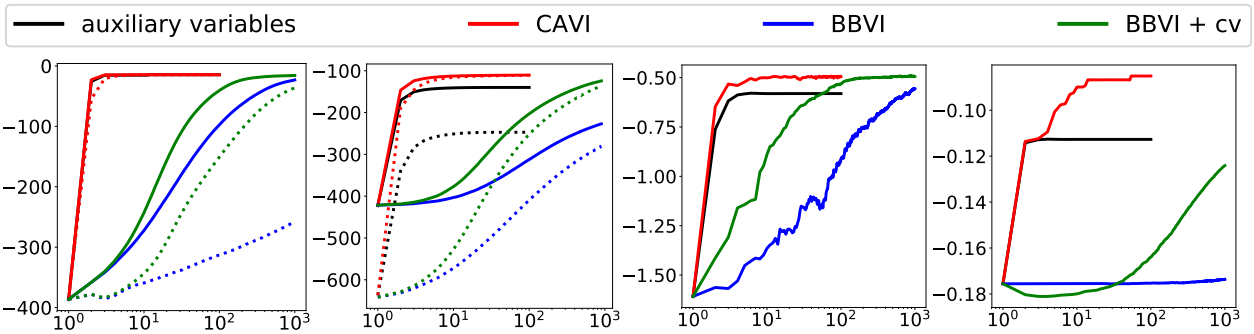
where $u_{\mathcal{P}(j)}^{(m)}$ is defined same as before. It is easy to see Equation (4.21) would reduce to Equation (4.6) when z_i is binary.

4.5 Experiments

We compare the proposed algorithm with BBVI and auxiliary-variable coordinate methods on the three models described in Section 4.2. We use the same datasets as in the original papers, and evaluate their average test ELBOs using Monte Carlo sampling, as in Gan et al. [2015]. We find that our Monte Carlo CAVI method has multiple appealing advantages over the baseline approaches.

4.5.1 Text Data and Noisy-OR Relations

As in Section 3.6, we use the tiny 20 Newsgroups dataset to test the inference performance, and follow the same model structure that has 44 latent topic nodes spanned in two layers, and 100 observed token nodes. The edge weights are fixed to the values learned through the variational training algorithm discussed in Chapter 3, without the local model settings.



(a) newsgroups (noisy-OR) (b) MNIST (sigmoid) (c) countries (Probit) (d) NIPS (Probit)

Figure 4.9: Improvement of average test ELBO (y axis) over iterations (x axis) on four different datasets. Our CAVI algorithm always converges to values higher than or similar to those of the model-dependent auxiliary-variable methods. It also converges in a speed orders of magnitude faster than BBVI, whose learning rate has been tuned for the best performance on each dataset. (a): CAVI behaves more robustly than BBVI when the sampling budget drops from 10 (solid) to 2 (dotted). (b): CAVI behaves more robustly than the auxiliary-variable method when the initialization changes from the marginal prior for each node (solid) to 0.5 (dotted). (c): Only on this very small dataset, BBVI with control variate (decaying average baseline) converges within 1,000 iterations. (d): CAVI is clearly better than the other methods on this larger relational dataset.

Table 4.1: Test ELBO of noisy-OR topic model on tiny 20 Newsgroups dataset. *Left:* Even with a very small sample size, CAVI outperforms BBVI and is comparable to the auxiliary-variable method. *Right:* Damping helps the convergence of parallel CAVI.

METHOD	ELBO
auxiliary variables	-14.53
CAVI 2 samples	-14.53
CAVI 10 samples	-14.51
BBVI 2 samples	-249.9
BBVI 10 samples	-21.22
BBVI 2 samples + cv	-34.00
BBVI 10 samples + cv	-15.76

METHOD	ELBO
sequential CAVI	-14.51
parallel CAVI, no damping	-50.54
parallel CAVI, $\alpha = 0.5$	-14.49
parallel CAVI, $\alpha = 0.25$	-14.49
parallel CAVI, $\alpha = 0.1$	-14.50

Fast Convergence with Small Sample Sizes

We compare the performance of different variational methods on the test set, which contains 4,872 documents in total. As before, the variational distribution q is all initialized to 0.5. The trace plot of average ELBO is shown in Figure 4.9(a). Similar to the auxiliary-variable coordinate algorithm, CAVI converges in about 10 iterations. As provided in Table 4.1, the ELBO of the two methods at convergence are very close (CAVI -14.51 v.s auxiliary variables -14.53). When the sample size drops from 10 to 2, the convergence speed of CAVI only slows down slightly.

On the contrary, BBVI is still far from convergence even after 1,000 iterations, just getting an average test ELBO of -15.76 with the control variate, and -21.22 without. We believe the slow ELBO improvement of BBVI is largely due to the low probability area around the optimum point shown in the toy example of Figure 4.4, especially for the last few hundreds of iterations. In addition, the dotted lines show that BBVI is much more vulnerable to the change in sample size.

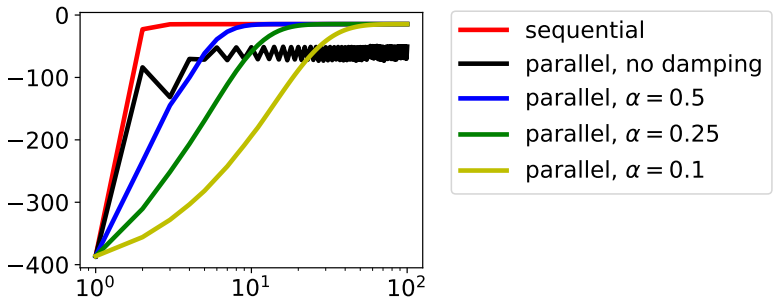


Figure 4.10: While the direct parallelization of CAVI (black) fails to converge, trails with damping all end up in good local optima similar to the sequential CAVI update (red). Larger damping rate α helps to converge faster.

Parallel CAVI Updates via Damping

Coordinate algorithms update the parameters one at a time, while holding all the others fixed. Comparing to gradient-based methods that change all the parameters together, this sequential setup naturally prohibits CAVI from being embarrassingly parallelized. Similar to Sun et al. [2013], we find reliable parallelization can be achieved through damping. As shown in Equation (4.22), the damping update sets the vector of logits τ at iteration $t+1$ as a linear combination of its value in the previous iteration t, and the parallel coordinate update for all variables:

$$\tau_{t+1} = (1-\alpha) \cdot \tau_t + \alpha \cdot \tau_{\text{parallel}}. \tag{4.22}$$

The parallel updates across all dimensions share the same set of Monte Carlo samples. Figure 4.10 illustrates that without damping, the ELBO of the parallel update (black line) oscillates and never converges. Damping helps avoid this problem, as shown by the blue, green and yellow curves. We find as a general rule, the ELBO at convergence is not sensitive to the damping rate α. It only affects the convergence speed slightly, ranging from 10 to 50 iterations in Figure 4.10.

Theoretically, under the same sample size, each CAVI update with damping needs twice the

Table 4.2: Test ELBO of sigmoid belief network on MNIST dataset. While initializing the variational distribution with prior marginals is always a better choice than using 0.5, our CAVI algorithm is much more robust to local optima.

METHOD	ELBO (PRIOR INIT)	ELBO (0.5 INIT)
auxiliary variables	-139.9	-247.0
CAVI	-110.4	-110.9
BBVI	-224.2	-276.6
BBVI + cv	-122.4	-133.0

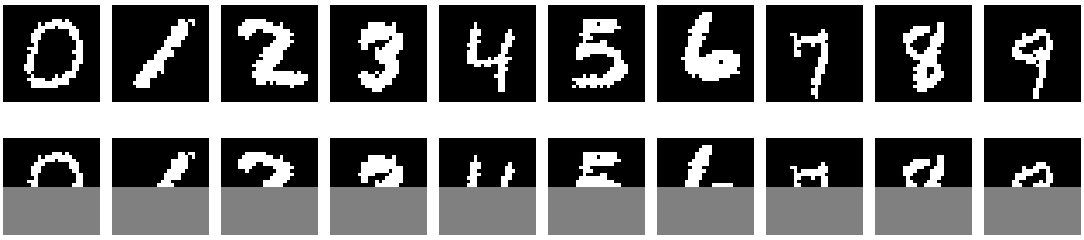
time of Rao-Blackwellized BBVI. That's because CAVI evaluates two sets of log densities in Equation (4.19), both $z_i = 1$ and $z_i = 0$, while BBVI only needs one of them in Equation (4.16), depending on the sample $z_i^{(m)}$. That said, since BBVI takes more iterations (and samples) to converge, its speed is much slower than CAVI in practice.

4.5.2 Image Data and Sigmoid Relations

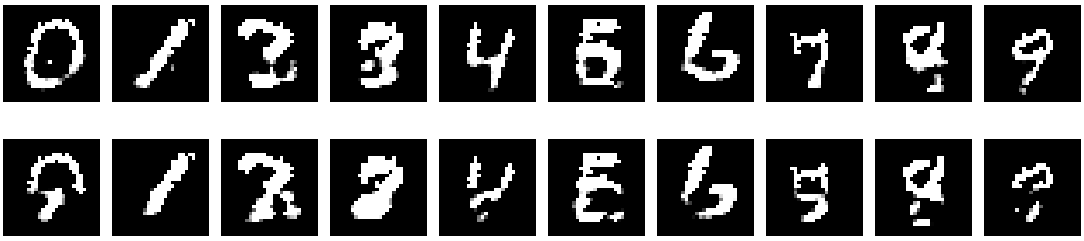
Following Gan et al. [2015], we build a fully-connected network with three layers. The two layers at the top have 100 nodes each, and the observed layer at bottom corresponds to the binarized images. We use the test set of MNIST, which contains 10,000 images each with 28×28 pixels. Edge weights of the network are learned from the training set through the public code of Gan et al. [2015] for coordinate-ascent variational training using the Pólya-Gamma trick.

CAVI is Less Vulnerable to Bad Initializations

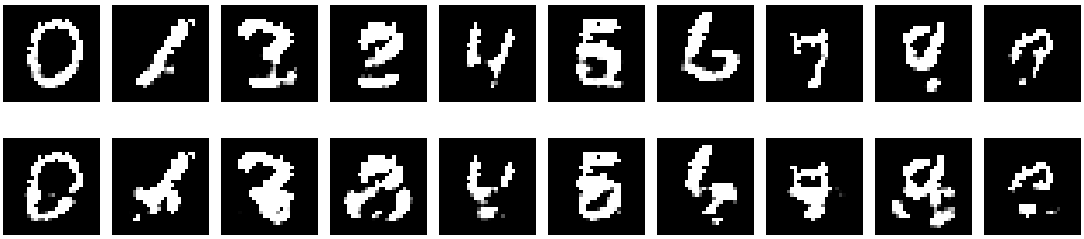
As presented in Figure 4.9(b), similar conclusions can be drawn for this model. Moreover, the auxiliary variable method performs very badly when q is uniformly initialized to 0.5. Changing the variable update order does not resolve this issue. It improves if we initialize q to be the marginal prior of each node, which we obtain via a Monte Carlo estimate over one



Input: bottom halves of the original images (top) are missing.



Initialization using prior marginals: CAVI (top) v.s BBVI (bottom).



Initialization using 0.5: CAVI (top) v.s BBVI (bottom).

Figure 4.11: Examples of MNIST digit completion. CAVI outperforms BBVI under both initialization settings.

million samples.

In contrast, CAVI is not as sensitive to initialization. As shown by the red lines in Figure 4.9(b), CAVI performs better than the auxiliary-variable method under both initialization strategies. Examples of digit completion in Figure 4.11 also illustrate this difference clearly.

We believe the reasons for the big difference in performance between CAVI and the auxiliary-variable method are two-fold. First, the auxiliary-variable objective is a lower bound of the ELBO, so it is expected that the result will be worse than CAVI, which optimizes the ELBO directly. Second, with more latent variables added in, the optimization surface becomes

Table 4.3: Test ELBO of latent-feature relational model on countries (left) and NIPS (right) datasets. While CAVI and BBVI with control variate reach about the same ELBO in the smaller countries dataset, their difference on the larger NIPS dataset is huge.

METHOD	ELBO	METHOD	ELBO
auxiliary variables	-0.581	auxiliary variables	-0.113
CAVI	-0.495	CAVI	-0.085
BBVI	-0.558	BBVI	-0.174
BBVI + cv	-0.494	BBVI + cv	-0.124

more complicated, so the data-augmented algorithm gets stuck in bad local optima more easily. We find the variance of the ELBOs that the auxiliary-variable method converges to are much larger than CAVI in repeated trials, where random orders of variable updates are used.

4.5.3 Link Data and Probit Relations

We test the performance on two datasets from the original paper of Miller et al. [2009]. The first one is the country dataset, which describes various relations (such as "accusation" and "economic aid") between 14 countries during 1950 to 1965 [Rummel, 1976]. In particular, we use the "conference" relation, which consists of symmetric connections indicating if two countries co-participate in any international conference. We set $D = 4$, and model parameters $w_d = 2, w_0 = -2, \rho = 0.5$ are selected through grid search. The features are initialized as the prior value ρ. As shown in Figure 4.9(c), on this very small model, BBVI with control variate finally reaches the same performance of CAVI after using over 10 times more iterations.

The second dataset is the NIPS co-authorship data by Globerson et al. [2007], where a link indicates two individuals being coauthors of a paper in one of the first 17 NIPS conferences. Following Miller et al. [2009] and Palla et al. [2012], we pick the 234 most connected authors,

and set $D = 10, w_d = 2, w_0 = -2, \rho = 0.1$. On this larger dataset, the advantage of CAVI over the auxiliary variable method and BBVI is very obvious. See Figure 4.9(d).

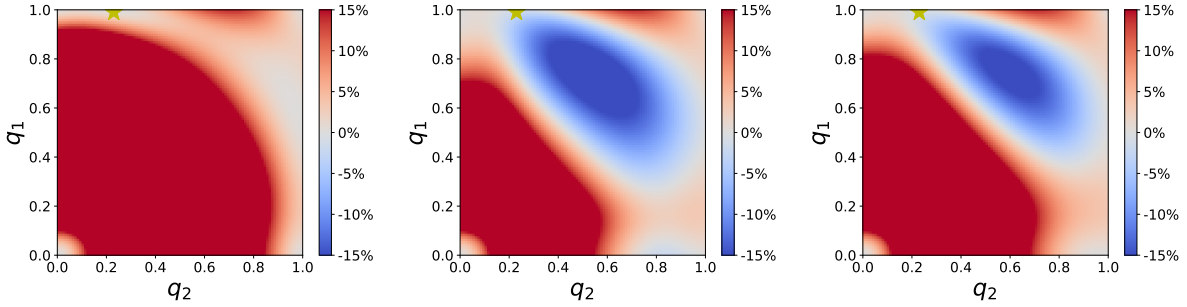
4.6 Discussion

We have developed a Monte Carlo variational inference framework applicable to any probabilistic model with discrete latent variables. The proposed method converges much faster than BBVI, and is less sensitive to the sample size and initialization. Relative to model-specific auxiliary bounds, our Monte Carlo CAVI algorithm directly optimizes a tighter likelihood bound, and is more robust to initialization in spite of being simpler to derive and implement.

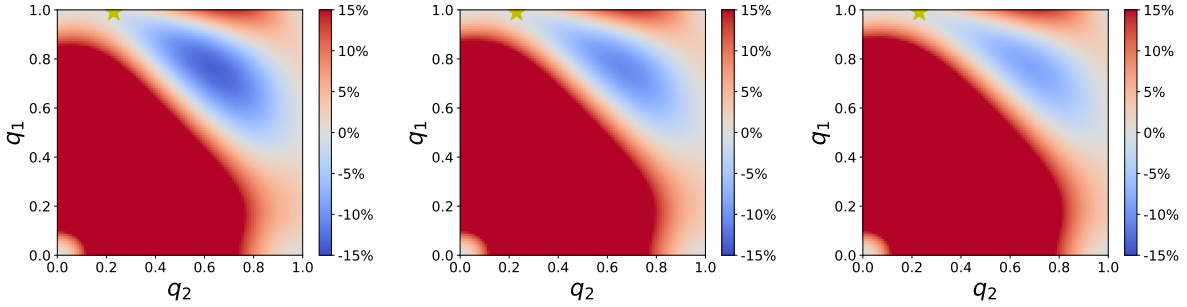
While we have mainly been focusing on models with binary variables for simplicity, it is straightforward to apply the general discrete-variable update in Equation (4.21) to other model families. We believe Monte Carlo coordinate ascent updates provide a compelling alternative to previous black-box variational methods as a scalable inference engine for probabilistic programs.

4.A Expected ELBO Increase for BBVI and CAVI on the Toy Model

Another way to visualize the difference between BBVI and CAVI on the toy example in Figure 4.3 is through the ratio of ELBO change $(\text{ELBO}_{\text{new}} - \text{ELBO}_{\text{old}})/|\text{ELBO}_{\text{old}}|$ after one update of q. As shown in Figure 4.12, the problematic blue areas of BBVI that tend to decrease the ELBO still exist, looking smoother under this new metric than in Figure 4.4. In contrast, our CAVI algorithm always improves the ELBO on average, no matter how many samples are used.

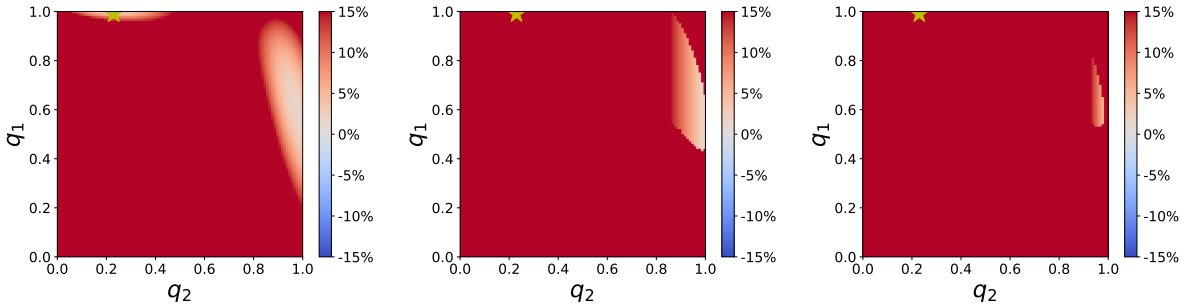


(a) analytic gradient (b) BBVI: 1 sample (c) BBVI: 2 samples

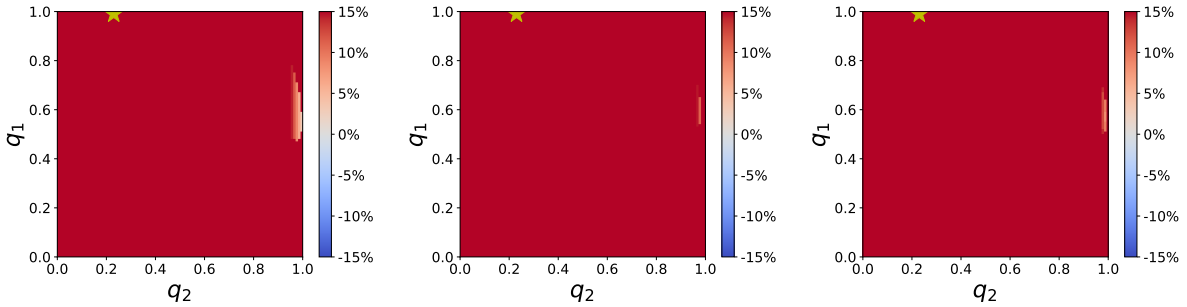


(d) BBVI: 3 samples (e) BBVI: 4 samples (f) BBVI: 5 samples

Figure 4.12: The expected proportion of ELBO increase after a gradient update of the logits of q, when the observation $x = 1$. The learning rate is 1.



(a) analytic coordinate update (b) CAVI: 1 sample (c) CAVI: 2 samples



(d) CAVI: 3 samples (e) CAVI: 4 samples (f) CAVI: 5 samples

Figure 4.13: The expected proportion of ELBO increase after a coordinate update of the logits of q, when the observation $x = 1$.

Chapter 5

Conclusion and Future Directions

Previous chapters developed efficient variational inference algorithms for image, text and network models. We now review the main contributions made in this thesis, and outline a few open areas for future research.

5.1 Summary of Methods and Contributions

Graphical models provide abundant tools to describe the relations within high-dimensional data. As their structures become more and more complex, the need for efficient inference techniques becomes a critical issue. We propose a family of variational inference algorithms for hierarchical models of images, text and social networks.

When only the observed data is given, we design the graphical models and the variational framework jointly to leverage the strength from both sides. In Chapter 2, we create a latent grid model that splits natural images into small regions, and ensures posterior dependencies between overlapping patches when pairing it with structural variational inference. A non-parametric upgrade using hierarchical Dirichlet process further captures the self similarities within each image, and enables novel clusters to be added dynamically during inference.

Chapter 3 considers the inference and learning problem when the probabilistic model is

already given and fixed. On deep Bayesian networks with noisy-OR relations, we develop a stochastic variational inference framework that efficiently updates the global edge weights using mini-batches of data, and infers the node activations via auxiliary bounds. Another primary contribution is the constrained family of variational bounds that greatly reduces the inference workload by leveraging the sparsity of observed word tokens of each document.

Finally, in Chapter 4 we design a general-purpose variational framework easily applicable to all discrete-variable models. By replacing the numerical evaluation of expectations with Monte Carlo sampling, our method scales linearly with the sample size no matter how complex the model structure is, and is straightforward to be integrated into probabilistic programming languages. Extensive experiments on recent models of text, images and networks show the advantages of our method against score-function gradients and hand-crafted auxiliary-variable methods, in terms of convergence speed, the number of samples required, the tightness of bounds, and the robustness to local optima.

5.2 Suggestions for Future Research

5.2.1 Multi-scale Patch-based Models for Natural Images

The HDP-GMM model designed in Chapter 2 mainly focuses on capturing the local dependencies between neighboring 8×8 patches. This would not suffice for tasks relying on long-range consistency in images, such as inpainting images with very large holes, or deblurring images taken with severe motions. Simply increasing the patch size is not a scalable solution, because the shape of each covariance matrix would boost quadrupally, let alone the potential increase in the number of clusters needed.

A promising direction then is to model the images at multiple scales, so that patches in the coarser scales correspond to larger regions of the original image [Papyan and Elad, 2015,

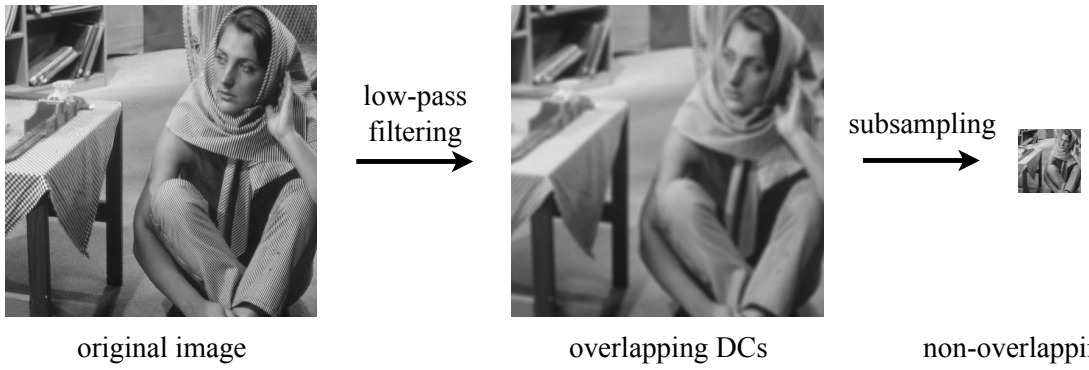


Figure 5.1: The DC offsets of non-overlapping patches can be viewed as a low-resolution image produced by filtering the original image with a uniform kernel and down sampling.

Shaham et al., 2019]. For example, one can build conditional GMMs to model the joint distribution of patches in different scales.

Another way to exploit the multi-scale idea is to improve the scalar Gaussian prior for the DC offsets in Chapter 2 with a better distribution that captures their joint density. As illustrated in Figure 5.1, the DC offsets of patches in each grid are equivalent to a subsampled version of the original image convolved with a uniform kernel. Thus to enforce the consistency at a coarser scale, we can use another GMM-grid prior to model the DC offset variables u_{mg} in Figure 2.1. A comparison is provided in Figure 5.2 where the multi-scale treatment nicely reduces artifacts for non-local patterns like the straight lines in the background. But an open challenge for multi-scale models is how to retain fine details in the restored images when long-range statistics are enforced.

5.2.2 Monte Carlo CAVI for Continuous-variable Models

In Chapter 4 we explore binary-valued models in depth, and provide Monte Carlo CAVI equations for more general models with discrete variables. One natural next step is to verify its effectiveness on concrete models in this regime, such as the ones for rating annotator expertise in crowd-sourcing applications [Welinder et al., 2010].

single scale: 28.58 dB multiple scales: **28.70** dB

Figure 5.2: Modeling the joint density of DC offsets enhances the long-range consistency of the restored "Lena" image (right). Compared to the single-scale denoising output (left), fewer artifacts are generated in areas such as the shoulder and the vertical bars in the background. Input noise level $\sigma = 50$.

A further direction is to build Monte Carlo CAVI algorithms for continuous models, or discrete ones with infinite numbers of discrete states. What we've already done does not extend straightforwardly to these types of distributions because it is impossible to enumerate the full range of the random variable. It would thus be interesting to see what ways of approximation or discretization could be used to overcome this issue. One idea could be to approximate these more difficult distributions with variational distribution in a parametric exponential family, such as Gaussian. Then the problem can reduce to approximating their finite vectors of sufficient statistics.

5.2.3 Efficient Implementation of Monte Carlo CAVI in PPLs

The Monte Carlo CAVI algorithm developed in Chapter 4 could in theory be integrated into existing probabilistic programming languages easily, because it only requires the same information needed to specify the probabilistic models. In practice, a nice property of our

algorithm is that the coordinate updates, including the parallelized version that incorporates damping, can be rewritten in a form similar to stochastic gradient updates. Therefore, it can be nicely plugged into PPLs built on top of modern deep-learning libraries, in a similar way that BBVI is implemented there.

We have built an initial version of Monte Carlo CAVI in Edward, but its running speed is not very high due to the extra manipulation of TensorFlow graphs. Thus we think it's worth the research and engineering efforts to develop an efficient PPL implementation of this general-purpose algorithm, either on top of popular ones like Edward (TensorFlow Probability), Pyro and Gen, or through a new probabilistic programming language that caters to the specific computation requirements of stochastic coordinate updates.

Bibliography

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.

M. J. Beal and Z. Ghahramani. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 1(4):793–831, 2006.

M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Computer Graphics and Interactive Techniques*, pages 417–424, 2000.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(1):973–978, 2019.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

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

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

C. Conati, A. S. Gertner, K. VanLehn, and M. J. Druzdzel. On-line student modeling for coached problem solving using Bayesian networks. In *International Conference on User Modeling*, pages 231–242, 1997.

M. F. Cusumano-Towner, F. A. Saad, A. K. Lew, and V. K. Mansinghka. Gen: A general-purpose probabilistic programming system with programmable inference. In *Conference on Programming Language Design and Implementation*, pages 221–236, 2019.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207, 2008.

G. Doyle and C. Elkan. Accounting for burstiness in topic models. In *International Conference on Machine Learning*, pages 281–288, 2009.

M. Drton and M. H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Z. Gan, R. Henao, D. Carlson, and L. Carin. Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, pages 268–276, 2015.

D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995.

A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

V. Gogate and P. Domingos. Formula-based probabilistic inference. In *Conference on Uncertainty in Artificial Intelligence*, pages 210–219, 2010.

N. D. Goodman and A. Stuhlmüller. The Design and Implementation of Probabilistic Programming Languages. `http://dippl.org`, 2014.

P. Gopalan, W. Hao, D. M. Blei, and J. D. Storey. Scaling probabilistic models of genetic variation to millions of humans. *Nature genetics*, 48(12):1587, 2016.

P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

Y. Halpern and D. Sontag. Unsupervised learning of noisy-OR Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 272–281, 2013.

M. Henrion. Search-based methods to bound diagnostic probabilities in very large belief nets. In *Conference on Uncertainty in Artificial Intelligence*, pages 142–150, 1991.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2(3):247–302, 1988.

M. C. Hughes and E. B. Sudderth. Memoized online variational inference for Dirichlet process mixture models. In *Advances in Neural Information Processing Systems*, 2013.

M. C. Hughes, D. I. Kim, and E. B. Sudderth. Reliable and scalable variational inference for the hierarchical Dirichlet process. In *Artificial Intelligence and Statistics*, 2015.

T. S. Jaakkola and M. I. Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Conference on Computer Vision and Pattern Recognition*, pages 1169–1176, 2009.

G. Ji, M. C. Hughes, and E. B. Sudderth. From patches to images: a nonparametric generative model. In *International Conference on Machine Learning*, pages 1675–1683, 2017.

G. Ji, D. Cheng, H. Ning, C. Yuan, H. Zhou, L. Xiong, and E. B. Sudderth. Variational training for large-scale noisy-OR Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Image denoising with nonparametric hidden Markov trees. In *International Conference on Image Processing*, 2007.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

S. Kong and C. Fowlkes. Image reconstruction with predictive filter flow. *arXiv preprint arXiv:1811.11482*, 2018.

A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei. Automatic variational inference in Stan. In *Advances in Neural Information Processing Systems*, pages 568–576, 2015.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1):430–474, 2017.

J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *International Conference on Management of Data*, pages 1729–1744, 2015.

J. Liu, X. Ren, J. Shang, T. Cassidy, C. R. Voss, and J. Han. Representing documents via latent keyphrase inference. In *International Conference on World Wide Web*, pages 1057–1067, 2016.

R. Liu, J. Regier, N. Tripuraneni, M. I. Jordan, and J. McAuliffe. Rao-Blackwellized stochastic gradients for discrete distributions. In *International Conference on Machine Learning*, 2019.

R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the Dirichlet distribution. In *International Conference on Machine Learning*, pages 545–552, 2005.

J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision*, volume 29, pages 54–62, 2009.

S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

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 *International Conference on Computer Vision*, 2001.

K. Miller, M. I. Jordan, and T. L. Griffiths. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, pages 1276–1284, 2009.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

R. M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

J. Paisley, C. Wang, and D. M. Blei. The discrete infinite logistic normal distribution. *Bayesian Analysis*, 7(2):235–272, 2012a.

J. W. Paisley, D. M. Blei, and M. I. Jordan. Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012b.

K. Palla, D. A. Knowles, and Z. Ghahramani. An infinite latent attribute model for network data. In *International Conference on Machine Learning*, 2012.

V. Papyan and M. Elad. Multi-scale patch-based image restoration. *IEEE Transactions on image processing*, 25(1):249–261, 2015.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108(504): 1339–1349, 2013.

J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

D. Ritchie, P. Horsfall, and N. D. Goodman. Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*, 2016.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, 2005.

D. L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37(23):3385–3398, 1997.

R. J. Rummel. *Attributes of nations and behavior of nation dyads, 1950-1965*. Inter-university Consortium for Political Research, 1976.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

T. R. Shaham, T. Dekel, and T. Michaeli. Singan: Learning a generative model from a single natural image. In *International Conference on Computer Vision*, pages 4570–4580, 2019.

J. Shi, J. Chen, J. Zhu, S. Sun, Y. Luo, Y. Gu, and Y. Zhou. ZhuSuan: A library for Bayesian deep learning. *arXiv preprint arXiv:1709.05870*, 2017.

M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann, and G. F. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of Information in Medicine*, 30(4):241–255, 1991.

T. Šingliar and M. Hauskrecht. Noisy-OR component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.

A. Srivastava, A. B. Lee, E. P. Simoncelli, and S. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *Conference on Computer Vision and Pattern Recognition*, pages 2451–2458, 2013.

J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: Extraction and mining of academic social networks. In *International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.

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

M. K. Titsias and M. Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems*, pages 2638–2646, 2015.

D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

D. Tran, M. W. Hoffman, D. Moore, C. Suter, S. Vasudevan, and A. Radul. Simple, distributed, and accelerated probabilistic programming. In *Advances in Neural Information Processing Systems*, pages 7598–7609, 2018.

G. Tucker, A. Mnih, C. J. Maddison, J. Lawson, and J. Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

C. Wang, W. Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *International Conference on Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

D. Wingate and T. Weber. Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*, 2013.

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

J. Yang and T. Huang. Image super-resolution: Historical overview and future challenges. *Super-resolution imaging*, pages 20–34, 2010.

L. Ye, A. Beskos, M. De Iorio, and J. Hao. On the efficacy of Monte Carlo implementation of CAVI. *arXiv preprint arXiv:1905.03760*, 2019.

X. Yi and J. Allan. A comparative study of utilizing topic models for information retrieval. In *European Conference on Information Retrieval*, pages 29–41, 2009.

M. Zontak and M. Irani. Internal statistics of a single natural image. In *Conference on Computer Vision and Pattern Recognition*, pages 977–984, 2011.

D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486, 2011.

D. Zoran and Y. Weiss. Natural images, Gaussian mixtures and dead leaves. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2012.