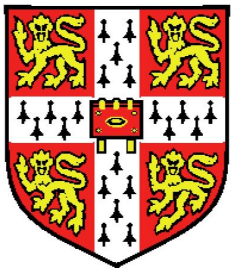
Submodular MAP Inference for Scalable Latent Feature Models



Colorado Reed

Department of Engineering

University of Cambridge

A thesis submitted for the degree of

Master of Philosophy

August 2013

Abstract

Latent feature models have become a keystone in the Bayesian nonparametrics community following the proposal of the Indian Buffet Process (IBP) [Griffiths and Ghahramani, 2006]: a stochastic process that describes a rich-get-richer probability on unbounded feature allocations. Inference for latent feature models is inherently difficult as the inference space grows exponentially with the size of the input data and number of latent features. This thesis shows a novel connection between inference with nonparametric latent feature models and submodular optimization. Specifically, this thesis shows that the log of the IBP distribution is submodular for each observation's feature assignments. As a result, we show that approximate MAP inference for a certain class of latent feature models can be phrased as a sequence of submodular maximization problems via a coordinate ascent framework. We further show how the maximization-expectation framework from Kurihara and Welling [2008] can be used in conjunction with our submodularity results to perform approximate MAP inference for latent feature models.

The submodularity property enables the use of scalable maximization algorithms that provide optimality guarantees when determining the MAP estimate for each observation's feature assignment. We focus our experiments and exploration on a nonnegative linear-Gaussian IBP model and outline how our results can be applied to other models. For the nonnegative linear-Gaussian model, we find that the submodular MAP framework scales linearly with the size of the input data, converges faster than sampling and variational techniques, and performs comparable in terms of predictive likelihood and L_2 error on a range of datasets. Furthermore, we show that the MAP results can be used to initialize sampling-based inference methods that perform better and converge to the target distribution faster than a random initialization.

Acknowledgements

Throughout this past year at Cambridge, I have had the benefit of working with talented, clever, and downright enjoyable researchers. In virtually any situation, if a person (be it a friend, colleague, cashier) said something along the lines of "I have this great idea, but I need someone to help me migrate this idea from 'idea-space' to 'reality-space,'" my adviser—Zoubin Ghahramani—would be this someone. Zoubin is an expert at working though ideas and turning them into tangible problems. From my own philosophical musings to highly specific technical ideas, Zoubin consistently helped me transform my thoughts into research questions, algorithms, code, and ultimately, this thesis. Thank you, Zoubin.

My friends and colleagues in the CBL lab provided a continuum of stimulating discussions and support. In particular, David Duvenaud, Yarin Gal, Roger Grosse, Creighton Heaukulani, James Lloyd, Alex Matthews, Rowan McAllister, Dan Roy, Christian Steinruecken, and Mark van der Wilk were always willing to help me work through challenging ideas. I hope that I returned the favor in some small way, and I look forward to future collaborations.

My year in Cambridge would not have been possible without support from the Winston Churchill Foundation of the United States. Peter Patrikis, the Executive Director of the foundation, is the antithesis of bureaucracy: obtaining travel funds, stipend payments, and completing administrative details was simple and straightforward (dare I say, enjoyable?). Thank you, Peter. In this regard, I would like to thank Kelly Thornburg, the Fellowship Director at the University of Iowa. Kelly introduced me to the Churchill scholarship and spent countless hours mentoring me through the process. Thank you, Kelly.

The love, support, and encouragement from my family fostered my academic pursuits. Though, my family would have been equally supportive had I chosen a life of spoken-word poetry, investment banking, street juggling, or medicine. Thank you for giving me the foundation to pursue my interests. And of course, Kaitlyn. Thank you for turning this European experience into a European adventure. It was a phenomenal year, and I'm glad that we experienced it together.

Contents

Chapter 1

Introduction

A data analyst—whether a biologist, physicist, statistician, or economist—makes assumptions about the data he observes in order to formalize a statistical analysis and interpret the results. For instance, a botanist may measure the petal width of one hundred flowers he collected in a field near his lab. The botanist could assume the field contains three distinct types of flowers, each of which has a distinct mean petal width and variance. By making these assumptions, the botanist can then formalize a statistical procedure to perform inference and test his hypothesis. These assumptions appear in both Bayesian and frequentist analyses and introduce human subjectivity into the analysis. Generally, the more assumptions an analyst makes about the data, the easier it is to perform inference. The trade-off, however, is that increasing the number of assumptions increases the human subjectivity present in the analysis, and if the injected subjectivity is incorrect, then our inference and understanding of the data could also be incorrect.

Nonparametric statistics attempts to reduce the degree of subjectivity needed to perform statistical inference. In classical terms, nonparametric statistics typically circumvents parametric assumptions by avoiding a complete specification of the likelihood, $p(\boldsymbol{X}|\theta)$ for some observations $\boldsymbol{X}$ and model parameters θ, see Wasserman [2006].[1] Bayesian nonparametric statistics also aims to reduce the degree of subjectivity needed for analysis while maintaining the well-defined Bayesian calculus for explicitly updating our assumptions based upon observed

[1] An alternate definition of "nonparametric statistics" focuses on the related concept of distribution-free tests, which is not the focus here.

data. Specifically, given a likelihood $p(\boldsymbol{X}|\theta)$ and prior $p(\theta)$, we update our posterior beliefs using Bayes' Rule:

$$p(\theta|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\theta)p(\theta)}{p(\boldsymbol{X})}. \tag{1.1}$$

Unlike frequentist nonparametrics, however, Bayesian nonparametrics requires full specification of the model likelihood, $p(\boldsymbol{X}|\theta)$ in order to update our beliefs. It is through the prior specification, $p(\theta)$, that we define our Bayesian analysis as "nonparametric." Namely, θ is viewed as an infinite dimensional parameter and $p(\theta)$ places positive support over this space, which is equivalent to specifying θ as a stochastic process. Different specifications of $p(\theta)$ yield different Bayesian nonparametric models with different characteristics, interpretations and posterior updates.

Returning to our botanist example, suppose the botanist returns to the same field year after year in order to collect additional flower samples. Would it be valid for the botanist to assume that there are exactly three types of flowers that will ever grow in this field? Probably not. What, then, should the botanist assume? Five? Ten? Twenty? In the Bayesian nonparametric formalism, the botanist can treat the number of flower types as a random, unbounded quantity and incorporate relatively vague assumptions via a nonparametric prior, e.g. observing three flower types is probable, observing one hundred flower types is less probable, and observing ten million flower types is virtually impossible.

In Bayesian modeling, it is common to model observed data as arising from unobserved (latent) categorical factors, e.g. our botanist models the observed petal width as arising from the latent flower type. The Dirichlet process [Ferguson, 1973] is the most widely used and studied Bayesian nonparametric prior for so called clustering or *partitioning* models, where each observed datum is associated with a single underlying factor. In other words, given N observations, e.g. $\{1, \ldots, 6\}$, a partitioning model places the observations into K mutually exclusive and jointly exhaustive sets called blocks, e.g. $\{\{2, 6\}, \{1, 3, 5\}, \{4\}\}$, where K is unbounded when using the Dirichlet process. The Dirichlet process can be used to define a distribution on the infinite-dimensional space of discrete distributions (or atomic probability measures), where the discrete states are usually interpreted

as the partition assignments. By sampling from the draw of a Dirichlet process, we obtain random partition labels that are exchangeable. So by specifying this generative process, we have defined a distribution on random partitions. An equivalent way of obtaining this distribution is to specify the predictive process for assigning an observation to a specific block, and then deriving the probability associated with any given partition. For the Dirichlet process, the predictive process is known as the Chinese Restaurant Process (CRP) [Aldous, 1985] and can be used to interpret the properties of the probability distribution derived from the Dirichlet process. See Broderick *et al.* [2012] for a detailed discussion.

The constraint that each observation can result from a single underlying factor is limiting: perhaps the petal widths that our botanist measured were also influenced by whether the flower was pollinated, whether there was a drought, whether the flower grew under a tree, etc. Assuming K such binary factors, it would take 2^K partitions to express all possible combinations of these factors, many of which would have overlap: (grew under tree), (grew under tree and picked during a drought), (grew under a tree and picked during a drought and was pollinated), etc. Of course, we could use probabilistic partitioning with only K sets rather than binary (hard) partitioning with 2^K set, but this has the possibly undesirable characteristic that belonging to one set with high probability is negatively correlated with belonging to any other set, e.g. a high probability of the flower being collected during a drought decreases the probability that it grew under a tree.

Extending the idea of a partition model such that observations arise from the interactions of multiple factors results in a multiple-membership or *feature allocation model*. Here, given N observations, e.g. $\{1, \ldots, 6\}$, a feature allocation model allows each observation to exist in any number of K sets, e.g. $\{\{2, 6\}, \{2, 5, 6\}, \{6\}, \{6\}\}$. Each observation can have any combination of features or no features at all, yielding 2^K possible feature assignments for each observation. Note that every partition is a feature allocation but the converse does not hold. In this sense, a feature allocation is a generalization of a partition.

The most widely used and studied Bayesian nonparametric prior for feature allocation models is the beta process [Griffiths and Ghahramani, 2006; Hjort, 1990; Thibaux and Jordan, 2007]. The beta process can be used to specify a

distribution on the infinite-dimensional space of discrete measures, where the discrete states are interpreted as binary features. Note the key difference between the Dirichlet process and the beta process: the Dirichlet process is defined over discrete *distributions* (which must integrate to unity), while the beta process is defined over discrete *measures* and has no integration constraints. In turn, assigning more probability mass to a given discrete state with the beta process does not remove probability mass from the other states. By sampling from a Bernoulli process with a beta process as its base measure, we obtain exchangeable random feature allocations, which are explained in §2.2. Like the Dirichlet process, the probability distribution on feature allocations obtained from this generative process can also be obtained from a predictive process known as the Indian Buffet Process (IBP). See Broderick *et al.* [2012] for a detailed discussion.

While feature allocation models can be more expressive than partition models, the size of the inference space for a binary feature allocation model with K features is exponentially larger than a partition model with K sets. That is, each observation can take 2^K different assignments yielding 2^{NK} different feature models for N observations. Inference then involves inferring the posterior distribution over this exponentially large space. Furthermore, when using a nonparametric prior, we must simultaneously infer the number of latent features K. When using a beta process, K is finite with probability one for finite datasets [Thibaux and Jordan, 2007]. But for practical computation purposes, the number of possible states, 2^{NK} for some fixed N and finite but variable K, is effectively infinite.

Most nonparametric feature allocation models use Markov Chain Monte Carlo (MCMC) sampling [Robert and Casella, 2004] to infer various parameters of interest, i.e. the latent feature assignments. The massively combinatorial size of the inference space often leads to slowly converging samplers that experience difficulty overcoming local optima in the posterior state space [Doshi-Velez *et al.*, 2009b]. A substantial research effort has attempted to overcome this problem through collapsed sampling techniques, particle filters, variational methods, and approximate Maximum A Posteriori (MAP) techniques for feature allocation models, c.f. §4.8. Nevertheless, modern nonparametric feature allocation models are still largely limited to operating with small datasets, see e.g. Palla *et al.* [2012].

In this thesis, we show a new approach for performing approximate inference

with a certain class of nonparametric feature allocation models. Specifically, we show that the log of the distribution defined by the IBP is submodular for each observation's feature assignments. Submodularity is a property of set functions that enables the use of scalable (i.e. greedy) algorithms that obtain constant-factor optimality guarantees on NP-hard combinatorial optimization problems, such as determining optimal feature assignments. As a result, we show that approximate MAP inference for a certain class of feature allocation models can be phrased as a sequence of submodular maximization problems via a coordinate-ascent framework. We empirically show that this framework allows feature allocation models to be applied to the largest datasets to date for a linear-Gaussian feature allocation model. Furthermore when used as an initialization for sampling techniques, the sampler outperforms all other inference techniques for linear-Gaussian feature allocation models on a range of diverse datasets. Finally, we outline how the submodular MAP inference technique can be applied to a number of feature allocation models.

1.1 Thesis Guide

Here we provide a bief summary of this thesis. Note that a subset of the work presented in this thesis was first presented in Reed and Ghahramani [2013].

- Chapter 2 provides the background material needed to understand the technical contributions of this thesis.
- In Chapter 3, we prove that the distribution defined by the IBP is log-submodular for each observation's feature assignments; we also show this property holds for the parametric analogue to the IBP.
- In Chapter 4, we show how the submodularity results of Chapter 2 in conjunction with the maximization-expectation framework from Kurihara and Welling [2008] can be used to perform approximate MAP inference with a nonnegative linear-Gaussian IBP model. This chapter provides detailed experimental sections that empirically characterizes our inference methodology.

- In Chapter 5, we outline how the submodularity property from Chapter 3 can be used for sparse matrix factorization models (a generalization of the model used in Chapter 4), models for network data, and models for general binary data.
- Chapter 6 provides a summary of the contributions of this thesis and poses a number directions for future research.

1.2 Notation

This section provides a reference list of nomenclature used in this thesis. Generally, we use the following conventions: boldface variables are matrices with (row, column) subscripts, a dot in the subscript indicates all elements of given dimension, and lowercase variables are scalars. A lowercase non-boldface variable with the same letter is used for the elements of a matrix, e.g. for matrix $\boldsymbol{M}$ the (i, j) element is m_{ij}.

$\boldsymbol{X}$: observed random variables

$\boldsymbol{Z}$: latent random variables of a probabilistic model

$\boldsymbol{A}$: latent model parameters of a probabilistic model

$\theta \in \Theta$: represents arbitrary parameters of a probabilistic model from some parameter space Θ

$q(\cdot)$: represents a variational distribution

$[N]$ represents the set of natural numbers up to N: $\{1, 2, \ldots, N\}$

Chapter 2

Background

In this chapter, we provide background material needed to understand the core contributions of this thesis. This chapter is self-contained but was not intended to be a tutorial for those completely unfamiliar with these topics. Therefore, at the end of each section we provide a list of tutorial-oriented resources on the specified subject.

2.1 Exchangeability

Exchangeability is a fundamental concept in Bayesian statistics that underlies virtually all Bayesian models [Orbanz and Roy, 2013]. Loosely stated, exchangeability is a specific assumption that can be made about data in order to formulate a statistical model and perform inference. Exchangeability encodes the assumption that the probability structure used to describe the data is invariant to permutations of some component of the data. For instance, an exchangeable sequence is an infinite sequence of random variables that satisfy

$$p(X_1, X_2, \ldots) = p(X_{\pi(1)}, X_{\pi(2)}, \ldots) \tag{2.1}$$

where π is a permutation of $\mathbb{N} = \{1, 2, \ldots\}$ and $X_i, i \in \mathbb{N}$ is an infinite sequence of random variables. In other words, we can shift around the *order* in which we observe the sequence without affecting the probability distribution for the sequence. This is equivalent to saying that a permutation of the random variables

are equal in distribution.

de Finetti's theorem states that an exchangeable sequence of random variables is a mixture of i.i.d. samples:

$$p(X_1, X_1, \ldots) = \int \prod_i^{\infty} Q_\theta(X_i)\nu(d\theta) \tag{2.2}$$

where Q_θ, for some random variable $\theta \in \Theta$, represents a family of conditional distributions, and ν is the distribution of Θ; ν is commonly referred to as the *de Finetti mixing measure*. By assuming exchangeability, de Finetti's theorem provides a framework for probabilistic modeling, where generally, we can let Θ be the space of measures and ν be some distribution over these measures. This structure leads to several useful nonparametric models. For instance, if we let Q_θ be some discrete probability distribution and ν be the Dirichlet process, then we obtain a nonparametric distribution of exchangeable partitions (clusters) described by the Chinese Restaurant Process, see Aldous [1985]. Exchangeability is present for all probabilistic models where some set of random variables are conditionally independent given some latent random variable, e.g. hierarchical models.

Throughout this thesis, we focus on a nonparametric distribution over exchangeable feature allocations described by a stochastic process known as the Indian Buffet Process (discussed below). Though not explicitly relevant for the contributions of this thesis, for completeness we note that the distribution described by the Indian Buffet Process can be obtained by integrating over a beta process de Finetti mixing measure with a Bernoulli process conditional distribution, see Thibaux and Jordan [2007] for details.

Learning resources: Foti and Williamson [2012], Orbanz and Roy [2013], and §4.2 of Bernardo and Smith [2009].

2.2 Feature Allocations

In this section, we introduce feature allocations and discuss them in the context of latent feature models that use an exchangeable nonparametric prior known as the Indian Buffet Process (IBP). This introduction to feature allocations follows

Broderick *et al.* [2013b] and Griffiths and Ghahramani [2011].

Given a set of integers $[N] := \{1, \ldots, N\}$, a *feature allocation*, denoted f_N is a set of subsets of $[N]$ where each element occurs in a finite number of subsets, termed *features*. For instance $f_{[3]} = \{\{2,3\}, \{3\}, \{\}, \{2\}, \{2\}\}$ is a feature allocation where the first feature is $\{2,3\}$. Notice that each element can occur in multiple features or zero features (element 1 does not occur in any features), and a feature itself can be an empty set. We denote an arbitrary feature by A_i, where in the above example $A_2 = \{3\}$. The integer elements that occupy the features are typically viewed as indices of observed data.

In this thesis, we are interested in nonparametric distributions on exchangeable feature allocations. From Broderick *et al.* [2013b]: let $\mathscr{F}_N$ represent the space of all feature allocations for $[N]$ and let F_N be a random element (a "random feature allocation") from $\mathscr{F}_N$. A random feature allocation is exchangeable if and only if F_N is equal in distribution to a permutation of the elements of the features. In other words, the probability distribution on the random feature allocation only depends on the number of elements in each feature: the particular elements within the feature do not matter. Mathematically, $F_N \overset{d}{=} \sigma(F_N)$, which states that F_N is equal in distribution to a permutation of the elements, where $\sigma(F_N)$ applies a permutation to the elements of each feature: $\sigma(F_N) = \{\sigma(A) : A \in F_N\}$ where $\sigma(A) = \{\sigma(i) : i \in A\}$.

The set representation of feature allocations provides a succinct and general representation for discussing feature allocations. Computationally, however, it is often beneficial to view feature allocations as binary matrices. Given a feature allocation, f_N, with K features, the binary matrix representation of f_N is an $N \times K$ matrix where the element in row n and column k is one if $n \in A_k$. A central difference between the set representation and the binary matrix representation is that the matrix representation implicitly includes an ordering of the features, which corresponds to a labeling of the features in the set representation. As Broderick *et al.* [2013b] noted, one must take care when defining this ordering as it is possible to violate the exchangeability of the features.

The IBP is a stochastic process that describes a nonparametric prior on exchangeable feature allocations. Griffiths and Ghahramani [2006] derived the distribution described by the IBP by placing independent beta priors on Bernoulli

generated entries of an $N \times K$ binary matrix $\boldsymbol{Z}$, marginalizing over the beta priors, and letting the number of features, K, go to infinity. Formally, the parametric generative model for $\boldsymbol{Z}$ is

$$\pi_k|\alpha,\beta \sim \text{beta}(\frac{\alpha\beta}{K},\beta) \tag{2.3}$$

$$z_{nk}|\pi_k \sim \text{Bernoulli}(\pi_k). \tag{2.4}$$

The beta distribution is conjugate to the Bernoulli, so integrating over all values of π_k can be done analytically yielding [Ghahramani *et al.*, 2007]:

$$P(\boldsymbol{Z}|\alpha,\beta) = \prod_{k=1}^{K} \frac{\Gamma(m_k + \frac{\alpha\beta}{K})\Gamma(N - m_k + \beta)\Gamma(\frac{\alpha\beta}{K} + \beta)}{\Gamma(N + \frac{\alpha\beta}{K} + \beta)\Gamma(\frac{\alpha\beta}{K})\Gamma(\beta)} \tag{2.5}$$

where $m_k = \sum_{i=1}^{N} Z_{nk}$ and $\Gamma(\cdot)$ is the gamma function. This is the two-parameter variant of the IBP (specified by parameters α and β); the original IBP derivation focused on the single parameter IBP, which sets $\beta = 1$. In taking the infinite limit $K \to \infty$, however, $P(\boldsymbol{Z})$ is zero for any particular $\boldsymbol{Z}$. Griffiths and Ghahramani [2006] therefore took the limit of an equivalence classes of binary matrices, $[\boldsymbol{Z}]_{\text{lof}}$, defined by the "left-order form" (lof) ordering of the columns and show that $P([\boldsymbol{Z}]_{\text{lof}})$ has a non-zero probability as K goes to infinity. The lof ordering arranges the columns of $\boldsymbol{Z}$ such that the binary values of the columns are non-increasing, where the first row is the most significant bit, see Figure 2.1. The lof equivalence classes correspond to choosing an equivalent lof labeling scheme in the set representation for feature allocations. By using the lof equivalence class, a combinatorial factor of

$$\frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \tag{2.6}$$

is introduced into Eq. 2.5 to account for the multiplicity of the equivalence class, where K_h represents the number of distinct features with binary representation h: to obtain h for a given feature, each feature is represented as a binary vector of length N where the elements within the feature are set to 1 and all other elements are zero, and the first index of the vector is the most significant bit. We refer to

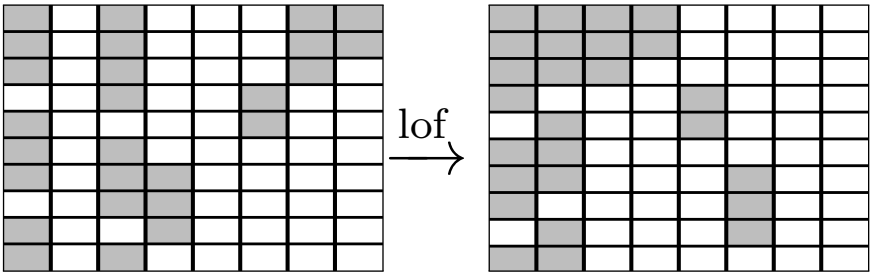


Figure 2.1: Example of a binary matrix (left) and its lof equivalence matrix (dark squares are 1, white squares are 0)—the columns are ordered by their non-increasing binary representation, where the first row is the most significant bit.

the binary representations of a unique feature as a "history." The combinatorial term from the equivalence class causes the probability of the equivalence class to remain well-defined and non-trivial as $K \to \infty$. The distribution for a lof equivalence class in the infinite limit is given by [Ghahramani *et al.*, 2007]:

$$P([\boldsymbol{Z}]_{\text{lof}}|\alpha,\beta) = \frac{(\alpha\beta)^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} e^{-\alpha\sum_{i=1}^{N}\frac{\beta}{\beta+i-1}} \prod_{k=1}^{K} \frac{\Gamma(m_k)\Gamma(N-m_k+\beta)}{\Gamma(N+\beta)}, \tag{2.7}$$

where K_+ is the number of columns with at least one non-zero entry—referred to as "active features." The IBP takes its name from a recursive culinary metaphor used to describe its predictive rule, and in turn, specify its distribution, Eq. 2.7.[1] The culinary metaphor is as follows:

1. The first customer (data index) enters an Indian restaurant and selects Poisson(α) dishes (feature labels) from a buffet

2. The *nth* customer ($n > 1$) chooses each of the previously selected dishes from the $1, \ldots, n-1$ customers with probability

$$\frac{m_{n-1,k}}{\beta + n - 1} \tag{2.8}$$

[1] As alluded to previously, it is also possible to obtain the IBP distribution by integrating over a Bernoulli process likelihood with a beta process de Finetti mixing measure. There are many equivalent techniques to derive the IBP probability distribution, see Griffiths and Ghahramani [2011].

and then samples a number of new features according to:

$$\text{Poisson}\left(\frac{\alpha\beta}{\beta+n-1}\right), \tag{2.9}$$

α is a mass parameter and β is a concentration parameter. This process is not exchangeable, and the probability obtained from its specification is not equivalent to Eq. 2.7; however, exchangeability is restored by examining the probability of the lof equivalence class of $\boldsymbol{Z}$.

The feature ordering scheme, or labeling scheme, for a feature allocation defines the equivalence class of the corresponding binary matrix. One computationally convenient alternative to the lof equivalence class is the "shifted" equivalence class first conjectured by Ding *et al.* [2010], shown to be the result of a uniformly random feature labeling scheme of observed features by Broderick *et al.* [2013b], and shown to be a valid equivalence class by Reed and Ghahramani [2013]. For a given $N \times K$ binary matrix $\boldsymbol{Z}$, the equivalence class $[\boldsymbol{Z}]_{\text{shift}}$ is obtained by shifting all-zero columns to the right of the non-zero columns while maintaining the non-zero column orderings, see Figure 2.2. This equivalence class multiplies the parametric feature probability, Eq. 2.5, by a combinatorial factor of $\binom{K}{K_+}$ to account for the multiplicity of the equivalence class. Similar to the lof equivalence class, this combinatorial term causes the probability of the shifted equivalence class to remain well-defined and non-trivial as $K \to \infty$. The probability for a shifted equivalence class in the infinite limit is [Reed and Ghahramani, 2013]:

$$P([\boldsymbol{Z}]_{\text{shift}}|\alpha,\beta) = \frac{(\alpha\beta)^{K_+}}{K_+!} e^{-\alpha\sum_{i=1}^{N}\frac{\beta}{\beta+i-1}} \prod_{k=1}^{K} \frac{\Gamma(m_k)\Gamma(N-m_k+\beta)}{\Gamma(N+\beta)}. \tag{2.10}$$

Where the only difference from the lof probability is that the histories term $\left(\prod_{h=1}^{2^N-1} K_h!\right)^{-1}$ has been replaced with a $K_+!^{-1}$ factor. For the lof, this factor penalizes $\boldsymbol{Z}$ matrices with identical columns. In the feature allocation perspective, this term penalizes features that are assigned to the exact same set of observations. The $K_+!$ term in the shifted equivalence class does not distinguish between identical and distinct columns of $\boldsymbol{Z}$, and in turn, does not penalize repeated feature assignments. In practice, it is difficult to engineer an inference setting in

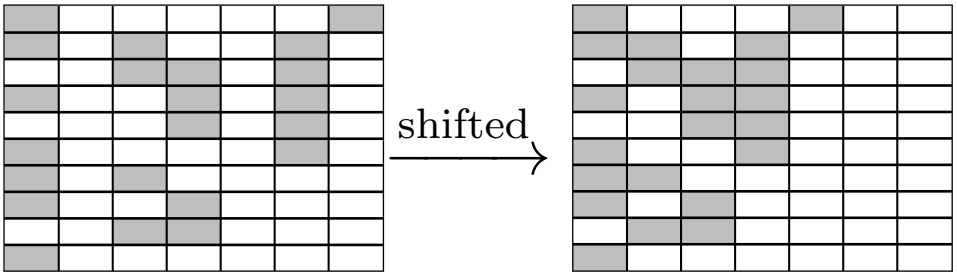


Figure 2.2: Example of a binary matrix (left) and its shifted equivalence matrix (dark squares are 1, white squares are 0)—placing the two all-zero columns anywhere in the matrix will yield the same equivalence matrix.

which two features are identical, as features often have associated parameters, i.e. model parameters, that distinguish the features even if they contain the same indices. These two equivalence class probabilities are proportional in the limit of large N as the probability of two columns being identical approaches 0.

Analytic inference for models with IBP components is intractable, so inference is accomplished through approximation techniques such as Markov Chain Monte Carlo (MCMC) sampling techniques. MCMC inference with IBP models is computationally expensive as its discrete state space has 2^{NK_+} possible assignments, where K_+ is unbounded but finite for finite datasets. In terms of computational ability, the number of possible feature assignments for a given dataset is effectively infinite, and as a result, samplers are often slow to converge and experience difficulty overcoming local optima [Doshi-Velez *et al.*, 2009b].

The IBP distribution has been used as a nonparametric feature allocation prior for a range of applications such as: unbounded independent component analysis models and factor analysis Knowles and Ghahramani [2007], nonparametric models of human similarity judgements [Navarro and Griffiths, 2007], protein interaction models [Krause and Wild, 2006], topic models [Williamson *et al.*, 2010], gene expression modeling [Knowles and Ghahramani, 2011], and network interaction models [Palla *et al.*, 2012]. However, due to computational constraints, all of the previously mentioned applications have been limited to small datasets. For example, Palla *et al.* [2012] applied their network model to a coauthorship network of only 234 authors, and Williamson *et al.* [2010] applied their topic model to a corpus with 2000 documents and a vocabulary of 1472 words—typical network

and topic modeling datasets are several orders of magnitude larger. Many authors have addressed the problem of scaling inference with latent feature models to larger datasets, we discuss this work in §4.8.

Learning resources: Broderick *et al.* [2013b] and Griffiths and Ghahramani [2011] provide a complementary discussion on feature allocations.

2.3 Variational Inference

Inference for probabilistic models typically focuses on computing the posterior distribution of latent random variables $\boldsymbol{Z}$ and model parameters $\boldsymbol{A}$ given observed data $\boldsymbol{X}$:

$$p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{Z})p(\boldsymbol{Z})p(\boldsymbol{A})}{p(\boldsymbol{X})}. \tag{2.11}$$

This computation is intractable for most probabilistic models of interest because

$$p(\boldsymbol{X}) = \int_{\boldsymbol{Z},\boldsymbol{A}} p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}) \tag{2.12}$$

is impossible to compute analytically. The central idea of variational inference is to approximate the posterior distribution, $p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})$, with a "variational distribution" $q(\boldsymbol{Z}, \boldsymbol{A})$ in order to perform an approximate posterior computation. In variational inference, the variational distribution is chosen to minimize the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior:

$$\mathrm{KL}(q(\boldsymbol{Z}, \boldsymbol{A})||p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})) = \mathbb{E}_q \left[\log \left(\frac{q(\boldsymbol{Z}, \boldsymbol{A})}{p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})} \right) \right] \tag{2.13}$$

where the expectation is computed with respect to the variational distribution. The KL-divergence is strictly nonnegative, implying that the global minimum (zero) can be obtained by setting $q(\boldsymbol{Z}, \boldsymbol{A}) = p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})$. Of course, determining the globally minimizing distribution returns us to our original problem of determining the posterior distribution, which is intractable. Therefore, we re-

strict $q(\boldsymbol{Z}, \boldsymbol{A})$ to a class of simpler distributions and find the distribution in this restricted class that minimizes the KL-divergence to the true posterior.

In Mean-Field Variational Bayes (MFVB) [Attias, 2000; Ghahramani and Beal, 2001], we restrict the distribution $q(\boldsymbol{Z}, \boldsymbol{A})$ to be factorized variational distributions:

$$q(\boldsymbol{Z}, \boldsymbol{A}) = q(\boldsymbol{Z})q(\boldsymbol{A}). \tag{2.14}$$

In this class of distributions, the latent random variables are independent and have their own variational distributions. While this class of distributions is very flexible (it can express any set of marginals of the latent random variables), it rarely contains the true posterior [Wang and Blei, 2012b].

Directly finding the variational distributions that minimizes the KL-divergence is difficult since we do not know the posterior distribution, i.e. the distribution that we are approximating. So instead, we maximize an objective function $\mathcal{L}(q)$ that is equivalent to minimizing the KL-divergence. Specifically, this objective function is a lowerbound on the log of the marginal likelihood:

$$\log\left(p(\boldsymbol{X})\right) = \mathcal{L}(q) + \mathrm{KL}(q(\boldsymbol{Z})q(\boldsymbol{A})||p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})) \tag{2.15}$$

$$\log\left(p(\boldsymbol{X})\right) \geq \mathcal{L}(q) \tag{2.16}$$

where

$$\mathcal{L}(q) = \mathbb{E}_q \log\left(\frac{p(\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{X})}{q(\boldsymbol{Z})q(\boldsymbol{A})}\right), \tag{2.17}$$

which is commonly referred to as the evidence lower bound (ELBO). It follows that $\mathcal{L}(q)$ is a lowerbound on $p(\boldsymbol{X})$ because $\mathrm{KL}(q(\boldsymbol{Z})q(\boldsymbol{A})||p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})) \geq 0$.

Via the calculus of variations, it is straightforward to show that setting $\partial\mathcal{L}(q)/\partial q = 0$ leads to optimal variational distributions that satisfy [Bishop, 2006]:

$$q^{\mathrm{OPT}}(\boldsymbol{Z}) \propto \exp\left(\mathbb{E}_{q(\boldsymbol{A})}\left[\log\left(p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{A})\right)\right]\right) \tag{2.18}$$

$$q^{\mathrm{OPT}}(\boldsymbol{A}) \propto \exp\left(\mathbb{E}_{q(\boldsymbol{Z})}\left[\log\left(p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{A})\right)\right]\right). \tag{2.19}$$

Note that $q^{\mathrm{OPT}}(\boldsymbol{Z})$ depends on $q(\boldsymbol{Z})$ and $q^{\mathrm{OPT}}(\boldsymbol{A})$ depends on $q(\boldsymbol{A})$, so it is

generally not possible to directly compute the optimal variational distributions. The optimization is therefore approximated via a coordinate ascent routine in which $q(\boldsymbol{Z})$ is optimized while $q(\boldsymbol{A})$ is held fixed and then $q(\boldsymbol{A})$ is optimized while $q(\boldsymbol{Z})$ is held fixed, etc. until the variational distributions converge. Under the coordinate ascent routine, the variational distributions are guaranteed to converge to a local optimum [Bishop, 2006].

Learning resources: Chapter 2 of Beal [2003]; Attias [2000].

2.4 Maximization-Expectation

Kurihara and Welling [2008] presented the ME algorithm: an inference algorithm that exchanges the expectation and maximization variables in the expectation-maximization (EM) algorithm [Dempster *et al.*, 1977]. Consider a general probabilistic model $p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{A})$, where $\boldsymbol{X}$ are the observed random variables (RVs), $\boldsymbol{Z}$ are the local latent RVs, and $\boldsymbol{A}$ are the global latent RVs. RVs are qualified as "local" if there is one RV for each observation, and RVs are "global" if the multiplicity of the parameters is constant or inferred from the data and are often referred to as "model parameters."

The EM algorithm can be viewed as a special case of MFVB that obtains MAP values of the global RVs by letting

$$q(\boldsymbol{A}) = \delta(\boldsymbol{A} - \boldsymbol{A}^*), \tag{2.20}$$

where $\delta(\cdot)$ is the delta function and $\boldsymbol{A}^*$ is the MAP assignment. The ME algorithm instead maximizes the local RVs $\boldsymbol{Z}$ and computes the expectation over the global RVs $\boldsymbol{A}$, which can be viewed as MFVB with

$$q(\boldsymbol{Z}) = \delta(\boldsymbol{Z} - \boldsymbol{Z}^*). \tag{2.21}$$

In the limit of large N, the ME algorithm recovers a Bayesian information criterion regularization term [Kurihara and Welling, 2008]. Also, maintaining a variational distribution over the global RVs retains the model selection ability of

MFVB, while using point estimates of the local RVs allows the use of efficient data structures and optimization techniques.

Learning resources: Kurihara and Welling [2008] provides a detailed discussion of ME inference.

2.5 Submodularity

Submodularity is a property of set functions that makes optimization of the set function tractable or approximable. Given ground set V and set function $f : 2^V \to \mathbb{R}$, f is *submodular* if for all $A \subseteq B \subseteq V$ and $e \in V \backslash B$:

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B), \tag{2.22}$$

which expresses a "diminishing returns" property, where the incremental benefit of element e diminishes as we include it in larger solution sets. Submodularity is desirable in discrete optimization because submodular functions are discrete analogues of convex functions and can be globally minimized in polynomial time [Lovász, 1983]. However, global submodular *maximization* is NP-hard, but submodularity often enables approximation bounds via greedy algorithms, cf. the resources at the end of this section.

Throughout this thesis, we will refer to the following two well-known properties of submodular functions.

Theorem 1. *Nonnegative linear combinations of submodular functions are submodular.*

Theorem 1 specifies one of the most useful properties of submodularity as it allows us to form complicated submodular functions by combining simple submodular functions. Theorem 1 is easily proven from the definition of submodularity, Eq. 2.22. Given a composite submodular function $r(A) = \sum_i \alpha_i t_i(A)$ where each function t_i is submodular and $\alpha_i \geq 0$ for all i. For set $A \subseteq B \subseteq V$ for base

set V, submodularity yields

$$\sum_i [\alpha_i t_i(A \cup \{e\}) - \alpha_i t_i(A)] \geq \sum_i [\alpha_i t_i(B \cup \{e\}) - \alpha_i t_i(B)] \tag{2.23}$$

where for an arbitrary but specific i we have

$$t_i(A \cup \{e\}) - t_i(A) \geq t_i(B \cup \{e\}) - t_i(B) \tag{2.24}$$

$$\alpha_i t_i(A \cup \{e\}) - \alpha_i t_i(A) \geq \alpha_i t_i(B \cup \{e\}) - \alpha_i t_i(B), \tag{2.25}$$

which is true since all t_i are submodular and α_i are nonnegative. It naturally follows that r is submodular. □

The next important submodular property involves *quadratic pseudo-Boolean* functions. A quadratic pseudo-Boolean function has the form

$$f(\boldsymbol{z}) = \boldsymbol{z}\boldsymbol{W}\boldsymbol{z}^T + \boldsymbol{z}\boldsymbol{\omega}^T + \text{const} \tag{2.26}$$

where $\boldsymbol{z}$ is a $1 \times K$ binary vector, $\boldsymbol{W}$ is a $K \times K$ quadratic weight matrix and $\boldsymbol{\omega}$ is a $1 \times K$ linear weight vector.

Theorem 2. *A quadratic pseudo-Boolean function with quadratic weight matrix* $\boldsymbol{W}$ *is submodular if and only if* $w_{ij} \leq 0$ *for all* i, j.

Theorem 2 is more esoteric than Theorem 1, but it is commonly cited in computer vision applications, cf. Kolmogorov and Rother [2007]. Its proof is simple: let $g(A)$, $A = \{i : \boldsymbol{z}_i = 1\}$ be the equivalent set function:

$$f(\boldsymbol{z}) = g(A) = \sum_{i \in A} \sum_{j \in A} w_{ij} + \sum_{i \in A} \omega_i + const. \tag{2.27}$$

and for $A \subseteq B \subseteq [K]$ and element $e \in [K] \setminus B$ submodularity yields,

$$\sum_{j \in A} w_{ej} + \omega_e \geq \sum_{j \in B} w_{ej} + \omega_e \tag{2.28}$$

$$\sum_{j \in A} w_{ej} \geq \sum_{j \in B} w_{ej} \tag{2.29}$$

$$-\sum_{j \in B \setminus A} w_{ej} \geq 0 \tag{2.30}$$

which is true for all $A \subseteq B \subseteq V = \{1, \ldots, K_+\}$ and elements $e \in V \setminus B$, proving the sufficiency of Theorem 2. The necessity of Theorem 2 is easily proven by contradiction: if any arbitrary but specific element w_{rs} in $\boldsymbol{W}$ was negative, then the inequality given in Eq. 2.30 would not hold when $r \in [K] \setminus B$ and $\{s\} \equiv B \setminus A$. □

Learning resources: Krause and Golovin [2012] and Chapter 5 of Krause [2008].

Chapter 3

Log-Submodular Feature Allocation Priors

In this chapter, we show that two forms of the IBP distribution and a parametric analogue (a finite beta-Bernoulli distribution) are log-submodular for each observation's feature assignment conditioned on the other observations' feature assignments. This implies that we can formulate approximate MAP inference for feature models as a sequence of submodular maximizations when all distributions, e.g. the likelihood, are also log-submodular as a function of each observation's feature assignment. We encapsulate the results of this chapter in the following theorems:

Theorem 3. *The IBP distribution with left-order-form equivalence classes is log-submodular for each observation's feature assignments.*

Theorem 4. *The IBP distribution with shifted equivalence classes is log-submodular for each observation's feature assignments.*

Theorem 5. *A parametric analogue of the IBP distribution (a finite beta-Bernoulli distribution) is log-submodular for each observation's feature assignments.*

We provide proofs and explanations of these theorems in this chapter.

In the next chapter, we show how the maximization-expectation framework from Kurihara and Welling [2008] can be used to perform approximate MAP inference with a nonnegative linear-Gaussian IBP model via a sequence of submodular maximizations. In Chapter 5, we outline additional techniques and models

for phrasing approximate MAP inference with feature models as a sequence of submodular maximizations.

3.1 Log-Submodularity of the IBP Distribution

The log of the two-parameter IBP distribution with left-order-form (lof) equivalence classes is (see §2.2):

$$\begin{aligned}\log\left(P([\boldsymbol{Z}]_{\text{lof}}|\alpha,\beta)\right) =& K_+\log\left(\alpha\beta\right) - \log\left(\prod_{h=1}^{2^N-1} K_h!\right) - \alpha\sum_{i=1}^{N}\frac{\beta}{\beta+i-1} \\ &+ \sum_{k=1}^{K_+}\log\left(\frac{\Gamma\left(m_k\right)\Gamma\left(N-m_k+\beta\right)}{\Gamma\left(N+\beta\right)}\right),\end{aligned} \tag{3.1}$$

where α, β are positive constants, $m_k = \sum_{n=1}^{N} z_{nk}$ are the number of non-zero indices in column k, and K_h are the number of features with binary representation h (see §2.2). The log of the two parameter IBP distribution with shifted equivalence classes is very similar:

$$\begin{aligned}\log\left(P([\boldsymbol{Z}]_{\text{shift}}|\alpha,\beta)\right) =& K_+\log\left(\alpha\beta\right) - \log\left(K_+!\right) - \alpha\sum_{i=1}^{N}\frac{\beta}{\beta+i-1} \\ &+ \sum_{k=1}^{K_+}\log\left(\frac{\Gamma\left(m_k\right)\Gamma\left(N-m_k+\beta\right)}{\Gamma\left(N+\beta\right)}\right),\end{aligned} \tag{3.2}$$

where the only difference from the lof equivalence class is that the history term is replaced with $K_+!$.

We begin by showing that the log of the shifted equivalence class IBP distribution is submodular for each observation. This proof is straightforward and will naturally lead us to the more complicated lof proof. Following these proofs, the reader may be curious whether the full joint log-IBP distribution is submodular, rather than just the conditional IBP distribution. In Appendix A.2 we show that the joint IBP distribution is not submodular.

3.1.1 Proving the Log of the IBP Distribution with Shifted Equivalence Classes is Submodular

We begin by showing that the shifted equivalence class IBP distribution as a function of each observation can be written as linear pseudo-Boolean function plus a regularization term that penalizes new features. After establishing this formulation, proving that it is submodular is straightforward.

3.1.1.1 Reformulating the Objective Function

Examining the log IBP distribution for each observation while fixing the other observations yields:

$$\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot}) = \log\left(\frac{\alpha\beta}{\Gamma\left(N+\beta\right)}\right)\kappa_n - \log\left((K_{+\backslash n}+\kappa_n)!\right) + \sum_{k=1}^{K_+}\log\left(\Gamma\left(m_{k\backslash n}+z_{nk}\right)\Gamma\left(N-m_{k\backslash n}-z_{nk}+\beta\right)\right) + const \quad (3.3)$$

where a "$\backslash n$" subscript indicates the given variable is determined after removing the n^{th} observation from $\boldsymbol{Z}$, e.g. $m_{k\backslash n} = \sum_{n'=1}^{N} z_{n'k} - z_{nk}$, and $\kappa_n = \sum_{k=1}^{K_+} \mathbf{1}_{\{m_{k\backslash n}=0\}} z_{nk}$ is the number of features unique to the n^{th} observation, and *const* absorbs the terms that do not depend on $\boldsymbol{Z}_{n\cdot}$. The purpose of this formulation is to separate the components of the IBP distribution that depend on $\boldsymbol{Z}_{n\cdot}$ from those that do not.

Next, we define the following auxiliary function:

$$\nu(z_{nk}) = \begin{cases} 0, & \text{if } m_{k\backslash n}=0 \text{ and } z_{nk}=0 \\ \log\left(\Gamma\left(m_{k\backslash n}+z_{nk}\right)\Gamma\left(N-m_{k\backslash n}-z_{nk}+\beta\right)\right) & \\ \qquad + \mathbf{1}_{\{m_{k\backslash n}=0\}} z_{nk} \log\left(\frac{\alpha\beta}{\Gamma(N+\beta)}\right), & \text{otherwise.} \end{cases}$$

This function incorporates all components of $\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot})$ except the $\log\left((K_{+\backslash n}+\kappa_n)!\right)$ term. Our goal is to write $\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot})$ as an inner product plus a regularization term. We do this by using the auxiliary function to form a vector $\mathbf{v}_n$ with com-

ponents:

$$v_{nk} = \nu(z_{nk} = 1) - \nu(z_{nk} = 0), \tag{3.4}$$

which lets us rewrite Eq. 3.3 as

$$\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot}) = \sum_{k=1}^{K_+} [z_{nk} v_{nk} + \nu(z_{nk} = 0)] - \log\left((K_{+\setminus n} + \kappa_n)!\right) + const. \tag{3.5}$$

Absorbing $\nu(z_{nk} = 0)$ into *const* for all $k \in [K_+]$ yields:

$$\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot}) = \boldsymbol{Z}_{n\cdot}\mathbf{v}_n - \log\left((K_{+\setminus n} + \kappa_n)!\right) + const, \tag{3.6}$$

which is a linear pseudo-Boolean function plus a regularizing term that penalizes new features.

3.1.1.2 Proving the Objective Function is Submodular

We prove the IBP distribution as a function of each observation, Eq. 3.6, is submodular by using the definition of submodularity given in §2.5. We use $\mathcal{G}_{\text{shift}}$ when treating Eq. 3.6 as a set function:

$$\mathcal{F}_{\text{shift}}(\boldsymbol{Z}_{n\cdot}) = \mathcal{G}_{\text{shift}}(A_n) = \sum_{a \in A_n} v_{na} - \log\left((K_{+\setminus n} + \kappa_{A_n})!\right) + const. \tag{3.7}$$

with $A_n = \{i : z_{ni} = 1\}$ and $\kappa_{A_n} = \sum_{k=1}^{K_+} \mathbf{1}_{\{m_{k\setminus n}=0\}} \mathbf{1}_{\{k \in A_n\}}$. To be submodular we must have

$$\mathcal{G}_{\text{shift}}(A_n \cup \{e\}) - \mathcal{G}_{\text{shift}}(A_n) \geq \mathcal{G}_{\text{shift}}(B_n \cup \{e\}) - \mathcal{G}_{\text{shift}}(B_n), \tag{3.8}$$

for all $A_n \subseteq B_n \subseteq [K_+]$ and elements $e \in [K_+] \setminus B_n$. From the definition of $\mathcal{G}_{\text{shift}}$, we need:

$$\begin{aligned} v_{ne} - \log\left((K_{+\setminus n} + \kappa_{A_n \cup \{e\}})!\right) + \log\left((K_{+\setminus n} + \kappa_{A_n})!\right) \geq \\ v_{ne} - \log\left((K_{+\setminus n} + \kappa_{B_n \cup \{e\}})!\right) + \log\left((K_{+\setminus n} + \kappa_{B_n})!\right), \end{aligned} \tag{3.9}$$

which simplifies to

$$\log\left(\frac{(K_{+\backslash n}+\kappa_{A_n})!}{(K_{+\backslash n}+\kappa_{A_n\cup\{e\}})!}\right) \geq \log\left(\frac{(K_{+\backslash n}+\kappa_{B_n})!}{(K_{+\backslash n}+\kappa_{B_n\cup\{e\}})!}\right). \tag{3.10}$$

Note that this simplification removes the linear terms from the inequality. This is a general property of submodular functions: linear set functions are submodular. Furthermore, note that models that have distributions that only linearly depend on $\boldsymbol{Z}$ through its selection of K_+ will demonstrate submodularity as the K_+ dependency can always be absorbed in a linear term. Eq. 3.10 has two cases:

1. $m_{e\backslash n} > 0$ so $\kappa_{B_n\cup\{e\}} = \kappa_{B_n}$ and $\kappa_{A_n\cup\{e\}} = \kappa_{A_n}$, yielding $0 \geq 0$ for Eq. 3.10, which is true for all $e \in [K_+] \setminus B_n$ and $A_n \subseteq B_n$.

2. $m_{e\backslash n} = 0$ so $\kappa_{B_n\cup\{e\}} = \kappa_{B_n} + 1$ and $\kappa_{A_n\cup\{e\}} = \kappa_{A_n} + 1$. Plugging this into Eq. 3.10 we have

$$\log\left(\frac{(K_{+\backslash n}+\kappa_{A_n})!}{(K_{+\backslash n}+\kappa_{A_n}+1)!}\right) \geq \log\left(\frac{(K_{+\backslash n}+\kappa_{B_n})!}{(K_{+\backslash n}+\kappa_{B_n}+1)!}\right). \tag{3.11}$$

   which simplifies to

$$\kappa_{B_n} \geq \kappa_{A_n}, \tag{3.12}$$

   which is again true for all $e \in [K_+] \setminus B_n$ and $A_n \subseteq B_n$.

As a result, the log of the IBP distribution for each observation's feature assignments is submodular. □

3.1.2 Proving the Log of the IBP Distribution with Left-Order-Form Equivalence Classes is Submodular

Similar to the previous subsection, here we show that the left-order-form equivalence (lof) class IBP distribution is submodular for each observation's feature assignment.

3.1.2.1 Reformulating the Objective Function

Because the only difference between the lof equivalence class IBP distribution and the shifted equivalence class IBP distribution is the histories term, a similar derivation to the one given in §3.1.1.1 results in the following reformulation of Eq. 3.1 (compare with Eq. 3.6):

$$\mathcal{F}_{\text{lof}}(\boldsymbol{Z}_{n\cdot}) = \boldsymbol{Z}_{n\cdot}\mathbf{v}_n - \Lambda(\boldsymbol{Z}_{n\cdot}) + const, \tag{3.13}$$

where $\Lambda(\boldsymbol{Z}_{n\cdot})$ (formally defined below) accounts for impact of $\boldsymbol{Z}_{n\cdot}$ on the histories term.

To define $\Lambda(\boldsymbol{Z}_{n\cdot})$, we must first specify some new notation. Let $h_{-n,k}$ denote the history of $\boldsymbol{Z}_{\cdot k}$ formed by removing z_{nk} from $\boldsymbol{Z}_{\cdot k}$. Furthermore define the collection of sets $S_n = \{s_{ni}\}$ for $i = 1, \ldots, 2^{N-1}$, where $s_{ni} = \{k : h_{-n,k} = i\}$, which specifies the indices of features that have the same history after removing $\boldsymbol{Z}_{n\cdot}$. Note that a maximum of K_+ sets in S_n will be nonempty and each feature index can only belong to one s set. For a given $\boldsymbol{Z}_{n\cdot}$, the histories term can be written as

$$\sum_{m=1}^{2^{N-1}} \log\left(K_m!\right) = \sum_{m=1}^{2^{N}-1} \log\left(\left(\sum_{k\in s_{nm}} z_{nk}\right)!\right) + \log\left(\left(\sum_{k\in s_{nm}} \bar{z}_{nk}\right)!\right). \tag{3.14}$$

For given $\boldsymbol{Z}_{n\cdot}$, S_n is fixed, so we can define $\Lambda(\boldsymbol{Z}_{n\cdot})$ strictly as a function of $\boldsymbol{Z}_{n\cdot}$ as follows:

$$\Lambda(\boldsymbol{Z}_{n\cdot}) = \sum_{m=1}^{2^{N}-1} \log\left(\left(\sum_{k\in s_{nm}} z_{nk}\right)!\right) + \log\left(\left(\sum_{k\in s_{nm}} \bar{z}_{nk}\right)!\right) \tag{3.15}$$

$$= \sum_{i:|s_{ni}|>1} \log\left(\left(\sum_{k\in s_{ni}} z_{nk}\right)!\right) + \log\left(\left(\sum_{k\in s_{ni}} \bar{z}_{nk}\right)!\right) \tag{3.16}$$

$$= \sum_{i:|s_{ni}|>1} \log\left(\left(\sum_{k\in s_{ni}} z_{nk}\right)!\left(|s_{ni}| - \sum_{k\in s_{ni}} z_{nk}\right)!\right) \tag{3.17}$$

$$= \sum_{i:|s_{ni}|>1} \log\left(\zeta_{ni}!\left(|s_{ni}| - \zeta_{ni}\right)!\right) \tag{3.18}$$

with $\zeta_{ni} = \sum_{k \in s_{ni}} z_{nk}$.

3.1.2.2 Proving the Objective Function is Submodular

Similar to §3.1.1.2, we prove the IBP distribution with the left-order-form equivalence classes is log submodular by using the definition of submodularity given in §2.5. We use $\mathcal{G}_{\text{lof}}$ when treating Eq. 3.6 as a set function:

$$\mathcal{F}_{\text{lof}}(\boldsymbol{Z}_{n\cdot}) = \mathcal{G}_{\text{lof}}(A_n) = \sum_{a \in A_n} v_{na} - \Lambda(A_n) + const, \tag{3.19}$$

with $A_n = \{i : z_{ni} = 1\}$, $\Lambda(A_n) = \sum_{i:|s_{ni}|>1} \log\left(\zeta_{A_n i}! \left(|s_{ni}| - \zeta_{A_n i}\right)!\right)$, and $\zeta_{A_n i} = |A_n \cap s_{ni}|$. To be submodular we must have

$$\mathcal{G}_{\text{lof}}(A_n \cup \{e\}) - \mathcal{G}_{\text{lof}}(A_n) \geq \mathcal{G}_{\text{lof}}(B_n \cup \{e\}) - \mathcal{G}_{\text{lof}}(B_n), \tag{3.20}$$

for all $A_n \subseteq B_n \subseteq [K_+]$ and elements $e \in [K_+] \setminus B_n$. Following the same steps as §3.1.1.2 leads us to the following inequality:

$$\Lambda(A_n) - \Lambda(A_n \cup \{e\}) \geq \Lambda(B_n) - \Lambda(B_n \cup \{e\}) \tag{3.21}$$

$$\sum_{i:|s_{ni}|>1} \log\left(\frac{\zeta_{A_n i}! \left(|s_{ni}| - \zeta_{A_n i}\right)!}{\zeta_{A_n \cup \{e\} i}! \left(|s_{ni}| - \zeta_{A_n \cup \{e\} i}\right)!}\right) \geq \sum_{i:|s_{ni}|>1} \log\left(\frac{\zeta_{B_n i}! \left(|s_{ni}| - \zeta_{B_n i}\right)!}{\zeta_{B_n \cup \{e\} i}! \left(|s_{ni}| - \zeta_{B_n \cup \{e\} i}\right)!}\right). \tag{3.22}$$

Without loss of generality, suppose the arbitrary but specific test element e belongs to set s_{nr} for some r. We have two cases:

1. $|s_{nr}| = 1$, then we have $\zeta_{A_n \cup \{e\} i} = \zeta_{A_n i}$ for all i (and likewise for B_n) yielding the inequality $0 \geq 0$

2. $|s_{nr}| > 1$ yields the inequality

$$\log\left(\frac{\zeta_{A_nr}!\,(|s_{nr}|-\zeta_{A_nr})!}{(\zeta_{A_nr}+1)!\,(|s_{nr}|-\zeta_{A_nr}-1)!}\right) \geq \log\left(\frac{\zeta_{B_nr}!\,(|s_{nr}|-\zeta_{B_nr})!}{(\zeta_{B_nr}+1)!\,(|s_{nr}|-\zeta_{B_nr}-1)!}\right)$$
$$\log\left(\frac{(|s_{nr}|-\zeta_{A_nr})}{(\zeta_{A_nr}+1)}\right) \geq \log\left(\frac{(|s_{nr}|-\zeta_{B_nr})}{(\zeta_{B_nr}+1)}\right)$$
$$(|s_{nr}|-\zeta_{A_nr})\,(\zeta_{B_nr}+1) \geq (|s_{nr}|-\zeta_{B_nr})\,(\zeta_{A_nr}+1)$$
$$\zeta_{B_nr}(|s_{nr}|+1) \geq \zeta_{A_nr}(|s_{nr}|+1)$$
$$\zeta_{B_nr} \geq \zeta_{A_nr} \tag{3.23}$$

which is true for all $A_n \subseteq B_n \subseteq [K_+]$ and elements $e \in [K_+] \setminus B_n$.

□

3.2 Log-Submodularity of a Parametric Beta-Bernoulli Distribution

Griffiths and Ghahramani [2006] derived the IBP distribution by placing independent beta priors on Bernoulli generated entries of an $N \times K$ binary matrix, marginalizing over the beta priors, and letting K go to infinity. Here we show that the parametric beta-Bernoulli distribution is log-submodular for each of the observation's feature assignments. The finite model specifies the following generative process

$$\pi_k \overset{\text{iid}}{\sim} \text{beta}(\frac{\beta\alpha}{K}, \beta) \tag{3.24}$$
$$z_{nk} \overset{\text{iid}}{\sim} \text{Bernoulli}(\pi_k) \tag{3.25}$$

where marginalizing over the π parameters leads to (see Ghahramani *et al.* [2007] for details):

$$\log\left(P(\boldsymbol{Z}|\alpha,\beta)\right) = \sum_{k=1}^{K} \log\left(\frac{\Gamma\left(m_k+\frac{\alpha\beta}{K}\right)\Gamma\left(N-m_k+\beta\right)\Gamma\left(\frac{\alpha\beta}{K}+\beta\right)}{\Gamma\left(N+\frac{\alpha\beta}{K}+\beta\right)\Gamma\left(\frac{\alpha\beta}{K}\right)\Gamma\left(\beta\right)}\right). \tag{3.26}$$

As a function of $\boldsymbol{Z}_{n\cdot}$, the log distribution is:

$$\mathcal{F}_{\text{finite}}(\boldsymbol{Z}) = \sum_{k=1}^{K} \log\left(\Gamma\left(m_{k\backslash n} + z_{nk} + \frac{\alpha\beta}{K}\right)\Gamma\left(N - m_{k\backslash n} - z_{nk} + \beta\right)\right) + const, \tag{3.27}$$

where the auxiliary function

$$\nu'(z_{nk}) = \log\left(\Gamma\left(m_{k\backslash n} + z_{nk} + \frac{\alpha\beta}{K}\right)\Gamma\left(N - m_{k\backslash n} - z_{nk} + \beta\right)\right) \tag{3.28}$$

lets us define the vector $\boldsymbol{\upsilon}_n$ of length K with components

$$\upsilon_{nk} = \nu'(z_{nk} = 1) - \nu'(z_{nk} = 0) \tag{3.29}$$

which allows us to rewrite the objective function as (see §3.1.1.1 for a similar derivation):

$$\mathcal{F}_{\text{finite}}(\boldsymbol{Z}_{n\cdot}) = \boldsymbol{Z}_{n\cdot}\boldsymbol{\upsilon}_n^T + const. \tag{3.30}$$

As we saw in §3.1.1.1, a linear pseudo-Boolean function (inner product) is trivially submodular. □

3.3 Discussion

In this chapter, we showed certain parametric and nonparametric feature allocation distributions are log-submodular for each observation's feature assignment. However, we have not discussed how to use these results to perform inference: the next two chapters focus on this problem. Briefly stated, the algorithmic frameworks formulate a coordinate ascent optimization routine that iteratively optimizes the $\boldsymbol{Z}_{n\cdot}$ assignments.

For instance, given a probabilistic model with joint probability, $p(\boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{A}|\boldsymbol{\theta}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\theta})p([\boldsymbol{Z}]|\boldsymbol{\theta})p(\boldsymbol{A}|\boldsymbol{\theta})$, for feature allocation $\boldsymbol{Z}$, model parameters $\boldsymbol{A}$, observations $\boldsymbol{X}$, and hyperparameters $\boldsymbol{\theta}$, we could iteratively maximize $\boldsymbol{Z}$ and $\boldsymbol{A}$ to obtain an approximate MAP solution. If $p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\theta})$ is log submodular, then

the results of this chapter in combination with Theorem 1 (which states that nonnegative linear combinations of submodular functions yields a submodular function) allows us to find approximate MAP assignments of $\boldsymbol{Z}$ via a sequence of submodular maximizations.

Chapter 4

Submodular MAP Inference for Nonnegative Linear-Gaussian Latent Feature Models

In this chapter, we show how the submodularity results of the previous chapter can be used to perform approximate MAP inference with a nonnegative linear-Gaussian IBP model. Specifically, we apply the maximization-expectation (ME) algorithm of Kurihara and Welling [2008] to a nonnegative linear-Gaussian IBP model and treat the evidence lower bound as an objective function for the latent feature assignments of an observation. As we show, this objective function is submodular and can be optimized using a simple greedy algorithm that provides a $\frac{1}{3}$ optimality guarantee. The ME algorithm and nonnegative linear-Gaussian IBP provide a concrete framework for performing submodular MAP inference with Bayesian nonparametric latent feature models, however, neither is essential for this development. In the next chapter, we discuss a broader class of models that display submodularity for each observation's latent feature assignments.

4.1 Nonnegative Linear-Gaussian IBP Model

We consider the following probabilistic model:

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{\theta}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \sigma_{\boldsymbol{X}}^2)p(\boldsymbol{A}|\sigma_{\boldsymbol{A}}^2)p([\boldsymbol{Z}]|\alpha) \tag{4.1}$$

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \sigma_{\boldsymbol{A}}^2) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{X}_{n\cdot}; \boldsymbol{Z}_{n\cdot}\boldsymbol{A}, \sigma_{\boldsymbol{A}}^2 I) \tag{4.2}$$

$$p(\boldsymbol{A}|0, \sigma_{\boldsymbol{A}}^2) = \prod_{k=1}^{K}\prod_{d=1}^{D} \mathcal{TN}(a_{kd}; 0, \sigma_{\boldsymbol{A}}^2) \tag{4.3}$$

$$P([\boldsymbol{Z}]|\alpha) = \frac{\alpha^{K_+}}{K_+!}\mathrm{e}^{-\alpha H_N}\prod_{k=1}^{K^+}\frac{(N-m_k)!(m_k-1)!}{N!}. \tag{4.4}$$

This is a nonnegative linear-Gaussian IBP model, where the prior over the latent factors, $p(\boldsymbol{A}|0, \sigma_{\boldsymbol{A}}^2)$, is a centred i.i.d. truncated Gaussian with nonnegative support, denoted $\mathcal{TN}$, see Appendix A.1 for a details of this distribution. Figure 4.1 provides a graphical illustration of the linear-Gaussian model. As we show, a nonnegative prior for the linear-Gaussian model yields a submodular maximization problem when optimizing $\boldsymbol{Z}$. We use a truncated Gaussian as it is conjugate to the Gaussian likelihood, but other nonnegative priors such as an exponential or beta prior can be plugged into this model with little change to the derivations below. For brevity we assume the hyperparameters, $\boldsymbol{\theta} = \{\alpha, \sigma_{\boldsymbol{A}}^2, \sigma_{\boldsymbol{X}}^2\}$, are fixed and discuss hyperparameter inference in Appendix A.3.2. For simplicity, we are using the single-parameter variant of the IBP prior (which is equivalent to the two-parameter IBP prior when $\beta = 1$) with the shifted equivalence classes, but any of the log-submodular priors specified in the previous chapter can be plugged into this model.

4.2 Evidence Lower Bound

In the ME framework, we approximate the true posterior distribution via a mean field variational Bayes (MFVB) assumption:

$$p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X}, \boldsymbol{\theta}) \approx q(\boldsymbol{A})\delta(\boldsymbol{Z} - \boldsymbol{Z}^*). \tag{4.5}$$

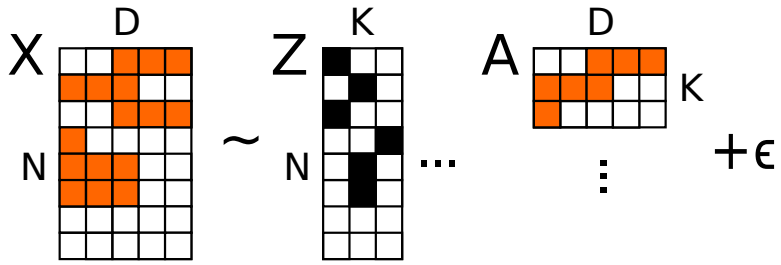


Figure 4.1: Graphical illustration of the linear-Gaussian model. The binary matrix $\boldsymbol{Z}$ linearly combines the factors $\boldsymbol{A}$ to form the observed data $\boldsymbol{X}$. This illustration is based on a similar illustration provided in Doshi-Velez [2009].

That is, we maintain a variational distribution over the latent factors $\boldsymbol{A}$ and optimize the latent features $\boldsymbol{Z}$. Given the MFVB constraint, we determine the variational distributions by minimizing the KL-divergence between the variational distributions and the true posterior, which is equivalent to maximizing a lower bound on the evidence, see §2.3 for a review of MFVB or Attias [2000] for a detailed discussion:

$$\begin{aligned} \log p(\boldsymbol{X}|\boldsymbol{\theta}) &= \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\theta})] + H[q] + D(q\|p) \\ &\geq \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\theta})] + H[q] \equiv \mathcal{L} \end{aligned} \tag{4.6}$$

where $H[q]$ is the entropy of variational distribution q and $D(q\|p)$ represents the KL-divergence between the variational distribution and the true posterior. The evidence lower bound (ELBO) for the nonnegative linear-Gaussian IBP model is:

$$\mathcal{L} = \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\theta})] + H[q] \tag{4.7}$$

$$\begin{aligned} &= \mathbb{E}_q[\log p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \sigma_{\boldsymbol{X}}^2)] + \mathbb{E}_q[\log p(\boldsymbol{A}|\sigma_{\boldsymbol{A}}^2)] + \mathbb{E}_q[\log p(\boldsymbol{Z}|\alpha)] \\ &\quad + H[q(\boldsymbol{A})] + H[q(\boldsymbol{Z})], \end{aligned} \tag{4.8}$$

which is simply the variational expectation over the log of the joint probability plus the entropy of the approximating distribution. The entropy of the delta function for a discrete random variable is zero, so we can remove this term. By inserting the appropriate probabilities and performing a bit of algebra (see

Appendix A.3.1 for details), the ELBO can be written as

$$\frac{1}{\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[-\frac{1}{2}\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T+\boldsymbol{Z}_{n\cdot}\boldsymbol{\xi}_{n\cdot}^T\right] - \log\left(K_+!\right)+\sum_{k=1}^{K^+}\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right)+\sum_{k=1}^{K}\eta_k+\text{const} \tag{4.9}$$

with

$$\xi_{nk}=\boldsymbol{\Phi}_{k\cdot}\boldsymbol{X}_{n\cdot}^T+\frac{1}{2}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}]^2-\mathbb{E}[a_{kd}^2]\right] \tag{4.10}$$

and

$$\eta_k=\frac{1}{2}\sum_{d=1}^{D}\left[-\log\left(\frac{\pi\sigma_{\boldsymbol{A}}^2}{2\alpha^{2/D}}\right)-\frac{\mathbb{E}[a_{kd}^2]}{\sigma_{\boldsymbol{A}}^2}+2H(q(a_{kd}))\right] \tag{4.11}$$

where

$$\boldsymbol{\Phi}_{k\cdot}=\left(\mathbb{E}\left[a_{k1}\right],\ldots,\mathbb{E}\left[a_{kD}\right]\right), \tag{4.12}$$

and all expectations are with respect to $q(\boldsymbol{A})$, which is defined in the next section. The ELBO should have $\boldsymbol{Z}^*$ instead of $\boldsymbol{Z}$ as we implicitly took the expectation with respect to $\delta(\boldsymbol{Z}-\boldsymbol{Z}^*)$, however, we will obtain $\boldsymbol{Z}^*$ by maximizing the ELBO with respect to $\boldsymbol{Z}$, so we abuse notation and let $\boldsymbol{Z}$ indicate $\boldsymbol{Z}^*$ in the context of evaluation.

By reformulating the ELBO, we see that it is the sum of N quadratic pseudo-Boolean functions that arise from the likelihood, plus a term that penalizes an increasing number of features ($-\log\left(K_+!\right)$) which arises from the IBP prior, plus a final term from the IBP and factor prior that encourages features to be active for a small or large number of observations (the $\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right)$ component) and have small-valued, high-entropy factors (the $-\frac{\mathbb{E}[a_{kd}^2]}{\sigma_{\boldsymbol{A}}^2}+2H(q(a_{kd}))$ component).

This ELBO demonstrates the classic "Occam's Razor" characteristic of Bayesian models: the variational distribution that maximizes the ELBO must describe the data well (else the quadratic pseudo-Boolean likelihood component will negatively

dominate the ELBO), while selecting a small number of informative features that reasonably match the properties of our prior (else the prior or entropy terms will negatively dominate the ELBO).

In §4.5 we show that obtaining $q(\boldsymbol{Z}) = \delta(\boldsymbol{Z} - \boldsymbol{Z}^*)$, i.e. maximizing this lower bound with respect to $\boldsymbol{Z}$, can be formulated as a submodular maximization problem.

4.3 Variational Factor Updates

Maximizing Eq. 4.9 with respect to $q(\boldsymbol{A})$ yields, see Appendix A.3.3,

$$q(\boldsymbol{A}) = \prod_{k=1}^{K} \prod_{d=1}^{D} \mathcal{TN}(a_{kd}; \tilde{\mu}_{kd}, \tilde{\sigma}_{kd}^2), \tag{4.13}$$

with parameter updates

$$\tilde{\mu}_{kd} = \rho_k \sum_{n=1}^{N} z_{nk}^* \Big(x_{nd} - \sum_{k' \neq k} z_{nk'}^* \mathbb{E}\left[a_{k'd}\right] \Big) \tag{4.14}$$

$$\tilde{\sigma}_{kd}^2 = \rho_k \sigma_X^2, \tag{4.15}$$

where $\rho_k = \left(m_k + \frac{\sigma_X^2}{\sigma_{\boldsymbol{A}}^2}\right)^{-1}$. These updates take $O(NK^2D)$, and the relevant moments are:

$$\mathbb{E}\left[a_{kd}\right] = \tilde{\mu}_{kd} + \tilde{\sigma}_{kd} \frac{\sqrt{2/\pi}}{\operatorname{erfcx}\left(\wp_{kd}\right)} \tag{4.16}$$

$$\mathbb{E}\left[a_{kd}^2\right] = \tilde{\mu}_{kd}^2 + \tilde{\sigma}_{kd}^2 + \tilde{\sigma}_{kd}\tilde{\mu}_{kd} \frac{\sqrt{2/\pi}}{\operatorname{erfcx}\left(\wp_{kd}\right)} \tag{4.17}$$

with $\wp_{kd} = -\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}$ and $\operatorname{erfcx}(y) = e^{y^2}(1 - \operatorname{erf}(y))$ representing the scaled complementary error function. Note that we are using $q(\boldsymbol{A})$ as shorthand for $q(\boldsymbol{A}|\tilde{\mu}_{kd}, \tilde{\sigma}_{kd})$.

The entropy of a truncated Gaussian is:

$$H(q(a_{kd})) = \frac{1}{2}\ln\frac{\pi e\tilde{\sigma}_{kd}^2}{2} + \ln\operatorname{erfc}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right) + \frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}}\sqrt{\frac{1}{2\pi}}\left(\operatorname{erfcx}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right)\right)^{-1}, \tag{4.18}$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function: $\operatorname{erfc}(y) = 1 - \operatorname{erf}(y)$.

4.4 Evidence Lower Bound in the Infinite Limit

The fastidious reader may have noticed that the IBP prior is defined for unbounded K—where only the active features affect the ELBO—but the entropy and prior terms for the latent factors are defined for all K. Here we show that the ELBO, Eq. 4.9, is well-defined in the limit $K \to \infty$; in fact, all instances of K are simply replaced by K_+. However, because we are using a variational approximation, a user must specify a maximum model complexity that K_+ cannot exceed. But unlike purely variational IBP methods [Doshi-Velez *et al.*, 2009b], the $q(\boldsymbol{Z})$ updates are not affected by inactive features.

We take this limit by breaking the ELBO into components $1, \ldots, K_+$ and $K_+ + 1, \ldots, K$ and note that when $m_k = 0$: $\tilde{\mu}_{kd} = 0$, $\tilde{\sigma}_{kd}^2 = \sigma_{\boldsymbol{A}}^2$, and $H(a_{kd}) = \frac{1}{2}\log\left(\frac{\pi e\sigma_{\boldsymbol{A}}^2}{2}\right)$. After some algebra, the ELBO becomes:

$$\psi_{K_+} + \frac{1}{2}\sum_{k=K_+ +1}^{K}\sum_{d=1}^{D}\left[-\log\left(\frac{\pi\sigma_{\boldsymbol{A}}^2}{2}\right) - \frac{\mathbb{E}[a_{kd}^2]}{\sigma_{\boldsymbol{A}}^2} + \log\left(\frac{\pi e\sigma_{\boldsymbol{A}}^2}{2}\right)\right] \tag{4.19}$$

where ψ_{K_+} is Eq. 4.9 but with K_+ replacing all K. From Eq. A.3, we see that $\mathbb{E}[a_{kd}^2] = \sigma_{\boldsymbol{A}}^2$ when $m_k = 0$, which causes all terms to cancel in Eq. 4.19 except ψ_{K_+}.

The ELBO remains well-defined because both the likelihood and IBP prior terms do not depend on inactive features, so for inactive features the KL-divergence between the posterior and variational distributions is simply the KL-divergence between $p(\boldsymbol{A})$ and $q(\boldsymbol{A})$. For inactive features, $p(\boldsymbol{A}) = q(\boldsymbol{A})$, and as a result, the KL-divergence is zero. This is a general result for the ME framework: as

long as the likelihood does not depend on inactive features and the priors are independent, the KL-divergence between the variational posterior and the true posterior will be zero for inactive features.

4.5 Feature Allocation Objective Function

Given $q(\boldsymbol{A})$, we compute MAP estimates of $\boldsymbol{Z}$ by maximizing the ELBO [Eq. 4.9] for each $n \in \{1, \ldots, N\}$ while holding constant all $n' \in \{1, \ldots, N\} \setminus n$. Decomposing Eq. 4.9 into terms that depend on $\boldsymbol{Z}_{n\cdot}$ and those that do not yields (see Appendix A.3.4):

$$\begin{aligned}
\mathcal{F}(\boldsymbol{Z}_{n\cdot}) =& -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot}\boldsymbol{\omega}_{n\cdot}^T + const \\
& - \log\left(\left(K_{+\setminus n} + \sum_{k=1}^{K_+}\left[\mathbf{1}_{\{m_{k\setminus n}=0\}}z_{nk}\right]\right)!\right) \\
\boldsymbol{\Phi}_{k\cdot} =& \left(\mathbb{E}\left[a_{k1}\right], \ldots, \mathbb{E}\left[a_{kD}\right]\right) \\
\omega_{nk} =& \frac{1}{\sigma_{\boldsymbol{X}}^2}\left(\boldsymbol{\Phi}_{k\cdot}\boldsymbol{X}_{n\cdot}^T + \frac{1}{2}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}]^2 - \mathbb{E}[a_{kd}^2]\right]\right) \\
& + \nu(z_{nk}=1) - \nu(z_{nk}=0) + \mathbf{1}_{\{m_{k\setminus n}=0\}}\eta_k,
\end{aligned} \tag{4.20}$$

which is a quadratic pseudo-Boolean function plus a term that penalizes K_+, where $\mathbf{1}_{\{\cdot\}}$ is the indicator function, a "$\setminus n$" subscript indicates the given variable is determined after removing the n^{th} row from $\boldsymbol{Z}$, and

$$\nu(z_{nk}) = \begin{cases} 0, & \text{if } m_{k\setminus n} = 0 \text{ and } z_{nk} = 0 \\ \log\left((N - m_{k\setminus n} - z_{nk})!(m_{k\setminus n} + z_{nk} - 1)!/N!\right), & \text{otherwise} \end{cases}$$

Proving this objective function is submodular is trivial given the results of the previous chapters. Namely, from Theorem 1 of §2.5, we know that nonnegative linear combinations of submodular functions are submodular. From Theorem 2 of §2.5, we also know that quadratic pseudo-Boolean functions are submodular if the quadratic weight matrix is nonpositive, which is the case for Eq. 4.20 since $\boldsymbol{\Phi}$ is

nonnegative. Finally, the regularization term $\log\left(\left(K_{+\backslash n}+\sum_{k=1}^{K_+}\left[\mathbf{1}_{\{m_{k\backslash n}=0\}}z_{nk}\right]\right)!\right)$ is from the log of the IBP prior, which as we showed in §3.1.1, is submodular. Therefore, Eq. 4.20 is a nonnegative linear combination of submodular functions, and by Theorem 1, it is submodular.

4.6 Finding the Optimal Feature Allocation

The submodular objective function, Eq. 4.20, is an unconstrained non-monotone submodular function. Feige *et al.* [2011] showed that an approximibility guarantee is NP-hard for this class of functions. However, Feige *et al.* [2011] also showed that a deterministic local-search algorithm obtains at least a $\frac{1}{3}-\frac{\epsilon}{K}$ approximation to the optimal solution for a ground set of size K and parameter ϵ, provided the submodular objective function is nonnegative. The local search algorithm queries the submodular function $O(\frac{1}{\epsilon}K^3\log(K))$ times for a ground set of size K, but as discussed below, this is a loose upper bound. In fact, a slight reformulation of their algorithm yields a $\frac{1}{3}-\epsilon$ approximation lower bound with an improved $O(\frac{1}{\epsilon}K^2\log(K))$ complexity. Recently, Buchbinder *et al.* [2012] proposed a linear greedy algorithm that obtained an improved $\frac{1}{3}$ approximation for nonnegative submodular functions using exactly $2K$ queries to the submodular function. Despite the improved complexity and performance bound, we find that the local-search algorithm performs better empirically. In the following subsections, we present these algorithms and discuss them in the context of the linear-Gaussian feature allocation objective function, Eq. 4.20.

4.6.1 Feige *et al.* [2011] Local Search Algorithm

For a submodular function $\mathcal{F}:2^V\rightarrow\mathbb{R}$ with ground set $V=[K]$, solution set $A\subseteq V$, and parameter ϵ, the local search (ls) unconstrained submodular maximization (USM) algorithm operates as described in Algorithm 1.

Algorithm 1 Deterministic Local Search USM Algorithm

1. *initialize*: let $A = \{\arg\max_{w \in V} \mathcal{F}(\{w\})\}$
2. *grow*: while there is an element $w \in V \setminus A$ s.t. $\mathcal{F}(A \cup \{w\}) > (1 + \frac{\epsilon}{K^2})\mathcal{F}(A)$: let $A := A \cup \{w\}$
3. *prune*: if there is an element $w \in A$ s.t. $\mathcal{F}(A \setminus \{w\}) > (1 + \frac{\epsilon}{K^2})\mathcal{F}(A)$: let $A := A \setminus \{w\}$, goto 2.
4. *return*: maximum of $\mathcal{F}(A)$ and $\mathcal{F}(V \setminus A)$.

The ls-algorithm makes $O(\frac{1}{\epsilon}K^3 \log(K))$ queries to the submodular function. This is a loose bound that occurs when all add/remove operations improve the objective function by a relative factor of exactly $\frac{\epsilon}{K^2}$, see Appendix A.4 for further discussion and §4.6.3 for an empirical characterization. The ls-algorithm obtains a solution that is greater than $\frac{1}{3}(1 - \frac{\epsilon}{K})\text{OPT}$, where OPT is the maximum value of $\mathcal{F}$.

In §3.1 of Feige *et al.* [2011], the authors prove the ls algorithm optimality bound and runtime by requiring that each grow/prune step increases the objective function by a relative factor of at least $1 + \frac{\epsilon}{K^2}$. If we instead require that the grow/prune steps increase the objective function by a relative factor of at least $1 + \frac{\epsilon}{K}$ their exact proofs remain valid, and the optimality bound becomes $\frac{1}{3} - \epsilon$ with improved worst case complexity $O(\frac{1}{\epsilon}K^2 \log(K))$. We use this variation of the ls algorithm throughout this thesis.

Since the submodular ELBO objective function, Eq. 4.20, is not strictly non-negative, we use its *normalized cost function* to interpret the ls-approximability guarantee: $\mathcal{F}(\boldsymbol{Z}_{n\cdot}) - \mathcal{F}_{n0}$, where $\mathcal{F}_{n0}$ is the minimum value of $\mathcal{F}(\boldsymbol{Z}_{n\cdot})$. Using the normalized cost function, we obtain the following optimality guarantee:

$$\mathcal{F}(\boldsymbol{Z}^{\text{ls}}_{n\cdot}) \geq \mathcal{F}_{n0} + \frac{1}{3}(1 - \epsilon)\left(\mathcal{F}(\boldsymbol{Z}^{*}_{n\cdot}) - \mathcal{F}_{n0}\right) \tag{4.21}$$

where the superscript "ls" denotes the solution from the ls-algorithm and an asterisk denotes the set that obtains the true maximum. This inequality states

that the ls-algorithm solution is guaranteed to perform better than the worst solution by an amount proportional to the difference between the optimal and worst solution. We emphasize, however, that this inequality does not provide an optimality guarantee for the global MAP solution.

Furthermore, we can set $\epsilon = 0$ to avoid computing the global minimum of the submodular function. This choice leads to an unbounded runtime according to the complexity bound from Feige *et al.* [2011]. However, we performed each of the experiments relevant to Figures 4.2-4.7 with both $\epsilon = 0$ and $\epsilon = 10^{-7}$ and found that the algorithms produced identical results in all cases. The reason this occured, as discussed further in Appendix A.4, is that adding or removing an element to the solution set either decreased the objective function or increased the objective function by several orders of magnitude more than the minimum value. From our experiments below and in §4.9, we conjecture that edge cases only result in contrived examples and choosing $\epsilon = 0$ is identical to choosing a sufficiently small, but non-zero, value. Therefore, unless noted otherwise, we use $\epsilon = 0$.

4.6.2 Buchbinder *et al.* [2012] Linear Greedy Algorithm

For a submodular function $\mathcal{F} : 2^V \to \mathbb{R}$ with ground set $V = [K]$, and sets C, E, the linear greedy (lg) algorithm operates as described in Algorithm 2.

Algorithm 2 Deterministic Linear Greedy USM Algorithm

1. *initialize*: let $C := V$, $E := \emptyset$
2. *loop*: for each element $w \in V$:
   if $\mathcal{F}(C \setminus \{w\}) - \mathcal{F}(C) > \mathcal{F}(E \cup \{w\}) - \mathcal{F}(E)$: remove w from C
   else: add w to E
3. *return*: C or E (they are equivalent)

This simple add/remove greedy algorithm makes exactly $2K$ calls to the submodular function, and Buchbinder *et al.* [2012] showed that this solution has a lower bound of $\frac{1}{3}$OPT. As before, the submodular ELBO objective function,

Eq. 4.20, is not strictly nonnegative so we must again use the normalized cost function to interpret the approximability guarantee:

$$\mathcal{F}(\boldsymbol{Z}^{\text{ls}}_{n\cdot}) \geq \mathcal{F}_{n0} + \frac{1}{3}\left(\mathcal{F}(\boldsymbol{Z}^{*}_{n\cdot}) - \mathcal{F}_{n0}\right), \tag{4.22}$$

where this approximability guarantee is tight and does not have the ϵ parameter like the local search algorithm.

Buchbinder *et al.* [2012] also propose a stochastic linear greedy algorithm that obtains an expected lower bound of $\frac{1}{2}$OPT. The algorithm is the same as the deterministic version except if adding an element to E increases the objective function by δ_E and removing and element from C increases the objective function by δ_C, then the element is added to E with probability $\frac{\delta_E}{\delta_C+\delta_E}$. In the event that δ_E is negative, the element is removed from C and vice-versa for a negative δ_C. Buchbinder *et al.* [2012] prove that $\delta_E + \delta_C \geq 0$, so both δ_E and δ_C will not be negative.

The optimality bounds for both the stochastic and deterministic algorithms do not depend on the order of traversal for the elements in V. From a number of informal tests, we found that both algorithms tend to perform better if the elements in V are greedily chosen such that at each step we choose the element to add/remove that results in the largest objective function increase. Locating this element takes $O(K)$ calls to the submodular objective function, and as a result, the complexity of the stochastic and deterministic ordered linear greedy algorithm is $O(K^2)$. We perform a formal comparison of these algorithms in the next subsection.

4.6.3 Submodular Maximization Experiments

We studied the empirical performance of the previously discussed submodular maximization algorithms in several contexts. We use the abbreviations listed in Table 4.1 to reference the submodular maximization algorithms.

In the first experiment, we generated $N = 500, D = 50$ high-noise data via

Table 4.1: Abbreviations for the submodular maximization algorithms discussed in this thesis for a submodular function of size K. Asterisks denote expected optimality guarantees for stochastic algorithms.

Abbreviation	Description	Number of submodular function queries	Optimality guarantee
ls	local search algorithm of Feige *et al.* [2011]	$O(\frac{1}{\epsilon}K^2\log(K))$	$\frac{1}{3}(1-\epsilon)$OPT
lg	deterministic linear greedy algorithm of Buchbinder *et al.* [2012]	$2K$	$\frac{1}{3}$OPT
lg-ord	deterministic linear greedy algorithm of Buchbinder *et al.* [2012] with greedy ordering	$O(K^2)$	$\frac{1}{3}$OPT
lg-sto	stochastic linear greedy algorithm of Buchbinder *et al.* [2012]	$2K$	$\frac{1}{2}$OPT$*$
lg-sto-ord	stochastic linear greedy algorithm of Buchbinder *et al.* [2012] with greedy ordering	$O(K^2)$	$\frac{1}{2}$OPT$*$
rand	randomly drawn solution	1	None

the following process:

$$a_{kd}|\sigma_{\boldsymbol{A}} \sim \mathcal{TN}(0, \sigma_{\boldsymbol{A}}) \tag{4.23}$$

$$\boldsymbol{Z}|\alpha \sim \text{IBP}(\alpha) \tag{4.24}$$

$$\boldsymbol{X}_{n\cdot}|\sigma_{\boldsymbol{X}} \sim \mathcal{N}(\boldsymbol{Z}_{n\cdot}\boldsymbol{A}, \sigma_{\boldsymbol{X}}\mathbb{I}) \tag{4.25}$$

with parameters $\sigma_{\boldsymbol{A}} = 1.0$, $\alpha = \frac{K}{10}$, and $\sigma_{\boldsymbol{X}} = 0.5$. We examined the performance of the submodular algorithms in optimizing the linear-Gaussian ELBO objective function for each $\boldsymbol{Z}_{n\cdot}$, Eq. 4.20, by comparing them to the brute-force optimal solution for $K = 2, 4, \ldots, 14$. In this experiment, we performed one optimization pass over all $\boldsymbol{Z}_{n\cdot}$ and optimized each $\boldsymbol{Z}_{n\cdot}$ independently of $\boldsymbol{Z}_{n'\cdot}$ for all $n' \neq n$ while holding $\boldsymbol{A}$ constant to either (case (i)) the true data-generating factors or (case (ii)) a random draw from the true generating distribution, $\mathcal{TN}(0, 1)$.

Figure 4.2 shows a boxplot of the fraction of the true maximum optimization value, $(F(S) - F_0)/(\text{OPT} - F_0)$ for solution set S, for each submodular maximization algorithm for case (i), and Figure 4.3 shows a similar boxplot for case (ii). Each boxplot was created from the results of one optimization pass over the 500 data instances for ten randomly generated input datasets for each K value, i.e. each K entry is composed of 5000 optimizations. The central mark on the displayed boxplots represents the median, the edges of the boxes are the 25th and 75th percentile, and the red crosses indicate the points that fall outside of the 25th and 75th percentile.

The boxplot characteristics for case (i) and case (ii) were similar: the ls and lg-ord consistently obtained over 99% of the optimum value across all K values, meaning that we could empirically replace the $\frac{1}{3}$ guarantee with a $\frac{99}{100}$ guarantee, though this is dependent on both the objective function and the input data. The other submodular maximization algorithms tend towards better solutions as K grows. This result is due to F_0 becoming increasingly negative as K increases, though for all K, these algorithms result in a larger performance variance and smaller performance mean than the ls and lg-ord algorithms. The random solution converges to a mean near 0.9 at $K = 4$ and maintains this mean through $K = 14$ with a decreasing variance as K grows, indicating that this particular set of optimization problems favor solutions near 0.9 across K values. Indeed, Figure 4.4

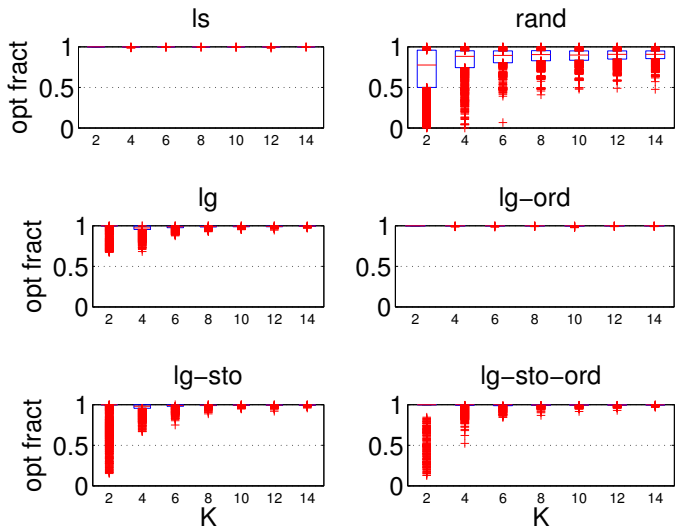


Figure 4.2: Fraction of the true maximum optimization value, $(F(S) - F_0)/(\text{OPT} - F_0)$ for solution set S, obtained for each of the five submodular optimization algorithms and a randomly chosen solution with $\boldsymbol{A}$ fixed to the true data-generating factors.

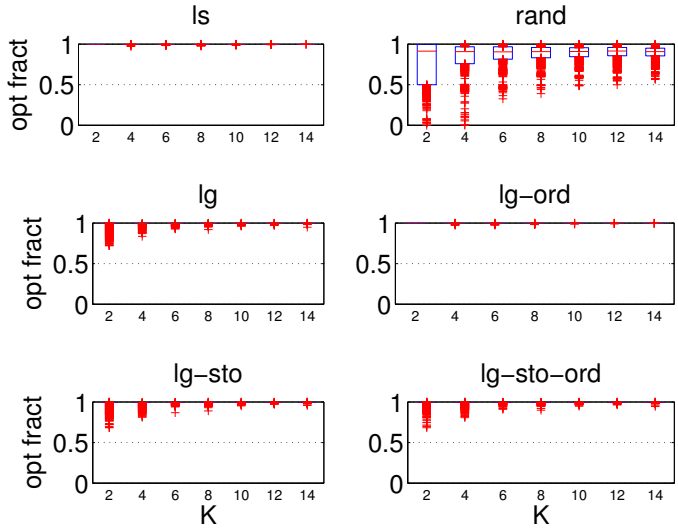


Figure 4.3: Fraction of the true maximum optimization value, $(F(S) - F_0)/(\text{OPT} - F_0)$ for solution set S, obtained for each of the five submodular optimization algorithms and a randomly chosen solution with $\boldsymbol{A}$ fixed to randomly drawn values from the true generating distribution.

shows the empirical density of $F(S)/\text{OPT}$ for each K value, and we see that the $F(S)/\text{OPT}$ densities tend towards a long tailed density with the majority of the mass focused between $[0.6, 1.0]$ with means between [0.8,0.95].

While case (i) and case (ii) tests produced similar results, the optimization fraction variances were always larger for case (i). This occured because F_0 tended to be less negative when $\boldsymbol{A}$ was set to its true value because the worst solution usually included all factors, which was less penalizing when some factors actually agreed with the data.

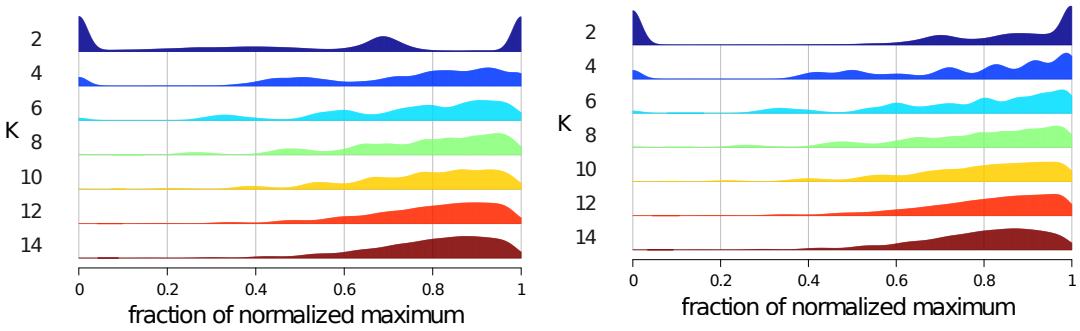


Figure 4.4: Empirical density of $F(S)/OPT$ for all solution sets S for various K values where $\boldsymbol{A}$ was fixed to the (**left**) the true generating factors and (**right**) a random draw from the true factor distribution. We determined the empirical density by sampling $\min(10^4, 2^K \times 500)$ $F(S)/OPT$ values from each of the ten trials at each K value. The density was estimated using the `kernel` function in the `stats` R library (a standard kernel density estimator with a Gaussian kernel with bandwidth 0.02).

We further studied the performance of the submodular maximization algorithms in a more-realistic MAP inference setting. Specifically, we generated $N = 500, D = 50$ noisy binary images by linearly combining a subset of the binary latent factors shown in Figure 4.5 (commonly referred to as the "Cambridge bars" factors), $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A} + \sigma_{\boldsymbol{X}}\mathbb{I}$. Each entry of the feature assignment matrix, $\boldsymbol{Z}$, was generated independently from a Bernoulli with probability 0.5, $z_{nk} \sim \text{Bernoulli}(0.5)$, and we fixed the noise at $\sigma_{\boldsymbol{X}} = 0.5$.

Rather than performing independent optimizations in a single pass over the data, we maintained the brute-force optimum feature assignments and updated $q(\boldsymbol{A})$ after each optimization and performed five iterations over the entire dataset

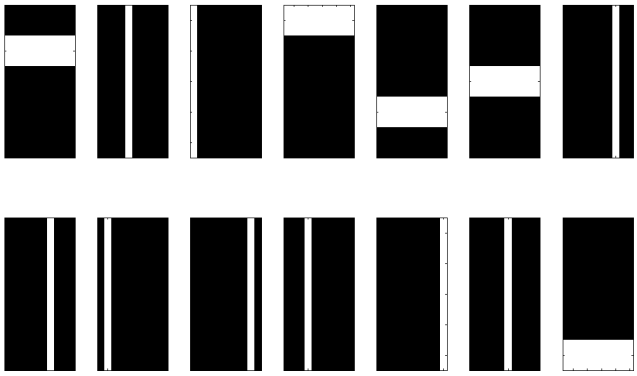


Figure 4.5: "Cambridge bars" binary latent factors. Note that the binary factors are mostly distinct but horizontal and vertical bars can overlap by exactly one square.

for each K value—this set-up resembled an actual submodular optimization inference procedure. The latent factors, $\boldsymbol{A}$, were initialized randomly from a centred truncated Gaussian with unit variance, and the feature assignments, $\boldsymbol{Z}$, were initialized from a random draw from Bernoulli(0.5). Figure 4.6 shows the boxplot from this experiment with each test performed with ten randomly generated datasets for each K value. These results again demonstrate that ls and lg-ord tend to outperform the other submodular maximization algorithms. Furthermore, the mean values of all submodular maximization algorithms (excluding the random solution) increased as K increased.

The solution variances for this experiment were larger than the solution variances for the previous experiments because the signal consisted of well-defined binary factors that caused the true maximum to be much larger than the previous experiments. In other words, the factors in the previous experiements closely resembled each other, causing misspecified solutions to be relatively close to the optimum. In this experiment, however, misspecifying the latent binary factors resulted in a larger penalty as the factors did not resemble each other. Figure 4.7 shows the empirical density of $F(S)/\text{OPT}$ for each K value for the Cambridge bars experiment. The variances of these densities were larger than the previous experiments because the data was composed of distinct binary factors. Solutions that included a subset of the true factors resulted in significantly better perfor-

mance than solutions that included incorrect factors, and in turn, the $F(S)/\text{OPT}$ density had a wider range of distinct values. We note, however, that as K increased, the density variance shrank and the mean converged to roughly 0.75—a similar characteristic as the previous experiments.

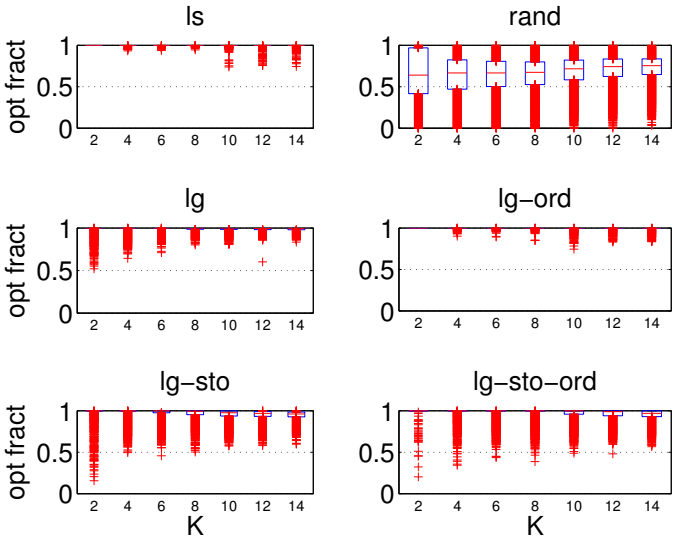


Figure 4.6: Fraction of the true maximum optimization value, $(F(S) - F_0)/(\text{OPT} - F_0)$ for solution set S, obtained for each of the five submodular optimization algorithms and a randomly chosen solution with $\boldsymbol{A}$ initialized to randomly drawn values from the true generating distribution.

Because of computational constraints, we are limited to small K values when comparing the submodular maximization algorithm results to the exact brute-force optimum. However, we can examine much larger K values by comparing the algorithms to each other rather than the optimal solution. In this experiment, we generated $N = 1000$, $D = 1000$ data from the linear-Gaussian model, where we randomly sampled K Cambridge bar factors from 10×100 binary images similar to those shown in Figure 4.5, sampled the feature assignments independently with 0.5 probability $z_{nk} \sim \text{Bernoulli}(0.5)$, and set the data noise to $\sigma_{\boldsymbol{X}} = 0.5$. Similar to our previous experiments, we independently optimized each $\boldsymbol{Z}_{n\cdot}$ in a single pass through the data while fixing $\boldsymbol{A}$ to either (case (i)) the true data generating factors or (case (ii)) a random draw from a centred truncated Gaussian with unit variance.

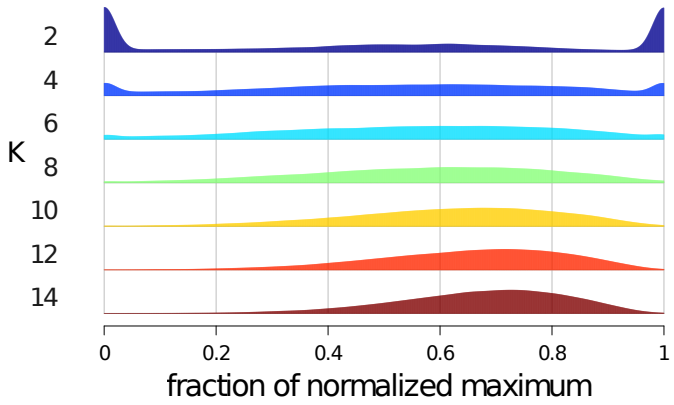


Figure 4.7: Empirical distribution of $F(S)/OPT$ for all solution sets S for various K for the Cambridge bars experiment. We determined the empirical density by sampling $\min(10^4, 2^K \times 500)$ $F(S)/OPT$ values from each of the ten trials at each K value. We estimated the density by using the `kernel` function in the `stats` R library (which uses standard kernel density estimation with a Gaussian kernel and a bandwidth of 0.02).

Figure 4.8 shows the fraction of maximizations that obtained the relative maximum value among the tested submodular maximization algorithms. The ls algorithm consistently outperformed the other algorithms in these experiements. All algorithms obtained the relative maximum value for $K \leq 35$ when $\boldsymbol{A}$ was fixed to a random draw from a truncated Gaussian. This occured because in all cases the relative maximum was the empty set: the misspecified fixed $\boldsymbol{A}$ did not explain the observed data. As K grew, however, certain factors explained the noise in the some of the data, which resulted in a nonempty relative maximum.

The previous experiments showed that (1) the ls algorithm tended to outperform the other submodular maximization algorithms on problems of a linear-Gaussian nature and (2) all algorithms substantially outperformed their worst case guarantees in practice. Iyer *et al.* [2013] also observed both of these characteristics on synthetic and real-world data that substantially differed from our linear-Gaussian data. While the results are data dependent, we note that the ls algorithm can empirically explore a larger fraction of the solution space by iteratively adding/removing elements from the solution set, while the other algorithms make a single greedy pass over the solution elements. The drawback, however, is that the ls algorithm has a larger upper bound on the number of submodular

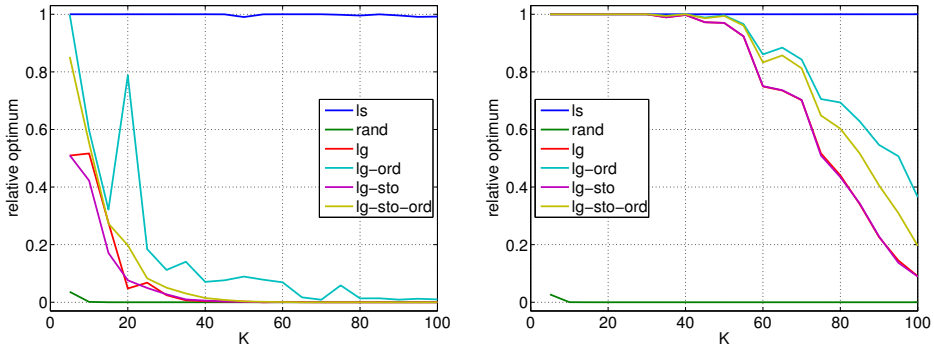


Figure 4.8: Fraction of maximizations that obtained the relative maximum value for the submodular maximization algorithms listed in Table 4.1, where for each K, the relative maximum is the maximum value obtained among all algorithms. The factors, $\boldsymbol{A}$, were fixed to the (**left**) true generating values and (**right**) a random sample from a zero-mean truncated Gaussian with unit variance. The fractions do not sum to one because multiple maximization algorithms can obtain the same maximum value.

queries. This upper bound is loose: in the next paragraph we show that it tends to scale subquadratically in practice.

We examined the empirical ls algorithm complexity by generating data from the nonnegative linear-Gaussian model with $N = 1000, D = 1000, \sigma_X = 1.0$ and performing 10 inference iterations for varying K values. By precomputing $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ and maintaining an auxiliary vector of K weights, we can evaluate Eq. 4.20 in constant time when adding/removing elements to the solution set. Figure 4.9 shows the number of constant-time queries to the submodular function vs the size of the ground set, K. For large K (roughly $K > 100$), the ls algorithm appeared to scale subquadratically, while for $K \leq 100$ a linear fit described the data well. NB: most latent feature modeling applications have $K \leq 100$. The other submodular optimization algorithms listed in Table 4.1 have a constant number of queries to the submodular function: $2K^2$ for the greedy lg algorithms and $2K$ for the non-greedy lg algorithms.

As discussed in Appendix A.4, the ls algorithm tended to scale better than its worst-case complexity because Feige *et al.* [2011] derived the complexity by assuming each add/remove step increases the objective function by the minimum possible amount. In practice, the objective function is either not improved or

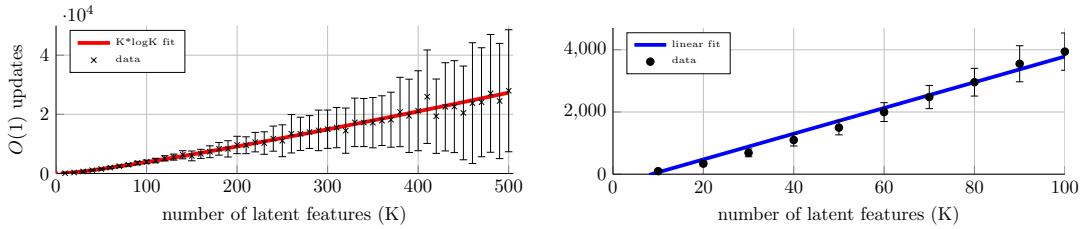


Figure 4.9: Number of $O(1)$ updates per ls optimization using data generated from the nonnegative linear-Gaussian model with $N = 1000, D = 1000, \sigma_{\boldsymbol{X}} = 1.0$. (**Left**) $K\log K$ fit to the entire data range, (**Right**) linear fit for $K \leq 100$.

is increased by an amount that is several orders of magnitude larger than the minimum possible amount.

4.7 MEIBP

Bringing together the components of the previous sections, we define the Maximization Expectation IBP (MEIBP) inference algorithm as specified in Algorithm 3. Note that we use the ls algorithm from Feige *et al.* [2011] as it outperformed the other submodular maximization algorithms in the previous section. By maintaining auxiliary matrices $\boldsymbol{Z}^T\boldsymbol{X}$ and $\boldsymbol{Z}^T\boldsymbol{Z}$, we can update $q(\boldsymbol{A})$ in $O(K_+^2 D)$ operations after each $\boldsymbol{Z}_{n\cdot}$ update, yielding a per-iteration complexity of $O(NK_+^2 D)$ for the $q(\boldsymbol{A})$ updates. By precomputing the quadratic pseudo-Boolean weight matrix $\boldsymbol{W}$ from Eq. 4.14, a $O(K_+^2 D)$ operation, we can maintain an auxiliary vector (aux in Algorithm 4) that keeps track of the net difference in the objective function when the k^{th} element is added/removed from the final solution set. In turn each add/remove operation with the ls algorithm performs $O(K_+)$ operations when finding the min/max element of the auxiliary vector, where the ls algorithm performs $O(\frac{1}{\epsilon}K_+\log(K_+))$ iterations: yielding a total per-iteration complexity of $O(NK_+^2(D + \frac{1}{\epsilon}\log(K_+)))$ for MEIBP.

Algorithm 3 The Maximization-Expectation IBP (MEIBP) inference algorithm for the nonnegative linear-Gaussian model.

input: observed data $\boldsymbol{X}$

initialize: set $\boldsymbol{Z}$ and $\boldsymbol{A}$ to random draws from their priors and determine $q(\boldsymbol{A})$ from Eq. 4.14

until convergence:

for each $n \in [N]$

$\boldsymbol{Z}_{n\cdot} \leftarrow$ ls-algorithm$(\boldsymbol{Z}, n, q(\boldsymbol{A}), \boldsymbol{X})$ (see Algorithm 4)

if $\boldsymbol{Z}_{n\cdot}$ changed:

update $q(\boldsymbol{A})$ from Eq. 4.14

return $\boldsymbol{Z}$, $q(\boldsymbol{A})$

Algorithm 4 The linear search algorithm from Feige *et al.* [2011] used for MEIBP inference.

input: feature assignment matrix $\boldsymbol{Z}$, data index n, observed data $\boldsymbol{X}$, and variational distribution $q(\boldsymbol{A})$

initialize: $\boldsymbol{W}$ and $\boldsymbol{\omega}_n$ from Eq. 4.20

aux $\leftarrow \boldsymbol{\omega}_n$

$K_{\text{final}} \leftarrow K_{+\backslash n}$

returnSet $\leftarrow \{\}$

if $\max_{k \in [K_+]} \left[\text{aux}_k - \mathbf{1}_{\{m_{k\backslash n}=0\}} \log\left(K_{\text{final}} + 1\right)\right] < 0$:

return returnSet

iterate until returnSet *does not change*

iterate until $\max_{k \in [K_+] \backslash \text{returnSet}} \left[\text{aux}_k - \mathbf{1}_{\{m_{k\backslash n}=0\}} \log\left(K_{\text{final}} + 1\right)\right] < \frac{\epsilon}{K_+}$:

newEl $\leftarrow \arg\max_{k \in [K_+] \backslash \text{returnSet}} \left[\text{aux}_k - \mathbf{1}_{\{m_{k\backslash n}=0\}} \log\left(K_{\text{final}} + 1\right)\right]$

returnSet $\leftarrow$ returnSet $\cup$ newEl

$K_{\text{final}} \leftarrow K_{\text{final}} + \mathbf{1}_{\{m_{\text{newEl}\backslash n}=0\}}$

aux $\leftarrow$ aux $+ \boldsymbol{W}_{\text{newEl}\cdot}$

iterate until $\min_{k \in \text{returnSet}} \left[\text{aux}_k - \mathbf{1}_{\{m_{k\backslash n}=0\}} \log\left(K_{\text{final}}\right)\right] > -\frac{\epsilon}{K_+}$

removeEl $\leftarrow \arg\min_{k \in \text{returnSet}} \left[\text{aux}_k - \mathbf{1}_{\{m_{k\backslash n}=0\}} \log\left(K_{\text{final}}\right)\right]$

returnSet $\leftarrow$ returnSet $\setminus$ removeEl

$K_{\text{final}} \leftarrow K_{\text{final}} - \mathbf{1}_{\{m_{\text{removeEl}\backslash n}=0\}}$

aux $\leftarrow$ aux - $\boldsymbol{W}_{\text{removeEl}\cdot}$

return: convert returnSet to binary vector of length K_+

4.8 Related Work: Scalable IBP Inference

Several proposals have been made for efficient inference with various latent feature models, with most techniques focusing on linear-Gaussian IBP models. Each of the techniques discussed below are iterative techniques, though the flavor of the iteration is different: each iteration for a sampler produces one (correlated) sample from the posterior of the model, while each iteration for the variational/MAP approaches optimizes some objective function that improves an approximation of the posterior. Table 4.3 summarizes the per-iteration complexity of the methods discussed below, but we remind the reader to take into account the differences between the inference methods when interpreting these complexities.

Doshi-Velez *et al.* [2009b] formulated a coordinate ascent variational inference technique for IBP models (VIBP); variational inference is discussed in §2.3. This method used the "stick breaking" formulation of the IBP, which maintained coupled beta-distributed priors on the entries of $\boldsymbol{Z}$—marginalizing these priors does not allow closed-form MFVB updates. Unlike MEIBP inference, maintaining the beta priors has the undesirable consequence that inactive features contribute to the evidence lower bound and must be ignored when updating the variational distributions. This was not a problem for Doshi-Velez *et al.* [2009b]'s finite variational IBP, which computed variational distributions for a linear-Gaussian likelihood with a parametric beta-Bernoulli prior on the latent features. The per-iteration complexity for both methods is $O(NK_+^2D)$, which is dominated by updating the variational distribution on the latent feature assignments.

Ding *et al.* [2010] used mixed expectation-propagation style updates with a mean field variational Bayes type of inference, termed "power-EP," to perform variational inference for a nonnegative linear-Gaussian IBP model (INMF). The expectation-propagation style updates are more complicated than standard mean field variational Bayes updates and have per-iteration complexity $O(N(K^3D + KD^2))$. Ding *et al.* [2010] motivated this framework by stating that the evidence lower bound of a linear-Gaussian likelihood with a truncated Gaussian prior on the latent factors is negative infinity. We note that this statement is only true if the variational distribution is fixed to be Gaussian, however the free-form variational distribution for their model is a truncated Gaussian, which has

a well-defined evidence lower bound.

Doshi-Velez and Ghahramani [2009] presented a linear-time "accelerated" Gibbs sampler for conjugate IBP models that effectively marginalized over the latent factors (AIBP). The per-iteration complexity was $O(N(K^2 + KD))$. This is comparable to the uncollapsed linear-Gaussian IBP sampler (UGibbs) that has per-iteration complexity $O(NDK^2)$ but does not marginalize over the latent factors, and as a result, takes longer to mix. In terms of both complexity and empirical performance, the accelerated Gibbs sampler is the most scalable sampling-based IBP inference technique currently available for linear-Gaussian IBP models. One constraint of the accelerated IBP is that it must be possible to analytically integrate the latent factor distribution out of the joint probability, which for instance, does not allow nonnegative priors on the latent factors.

Rai and Daume III [2011] introduced a beam-search heuristic for locating approximate MAP solutions to linear-Gaussian IBP models (BS-IBP). This heuristic sequentially adds a single data point to the model and determines the latent feature assignments by scoring all 2^{K_+} latent feature combinations. The scoring heuristic uses an estimate of the joint probability, $P(\boldsymbol{X}, \boldsymbol{Z})$ to score assignments, which evaluates the collapsed likelihood $P(\boldsymbol{X}|\boldsymbol{Z})$ for all 2^{K_+} possible assignments: an expensive $N^3(K_+ + D)$ operation, yielding a per-iteration complexity of $O(N^3(K_+ + D)2^{K_+})$.

Broderick *et al.* [2013a] showed that MAP estimates of a linear-Gaussian IBP model could be obtained by taking a zero-variance asymptotic limit of the model, $\sigma_{\boldsymbol{X}} \to 0$ with the IBP concentration parameter, α, set to

$$\alpha = \exp\left(\frac{-\lambda^2}{2\sigma_{\boldsymbol{X}}^2}\right) \tag{4.26}$$

for some constant $\lambda > 0$, resulting in a MAP objective function of the form

$$\arg\min_{K_+, \boldsymbol{Z}, \boldsymbol{A}} \left[\text{trace}\left((\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{A})^T(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{A})\right) + K_+\lambda^2\right], \tag{4.27}$$

which they optimize by greedily minimizing the objective function for each latent feature assignment, z_{nk} and then updating the latent factors to their conditional expectation, $E[\boldsymbol{A}|\boldsymbol{Z}, \boldsymbol{X}] = \boldsymbol{A} = (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{X}$. Features are added by greedily

checking whether one new feature reduces the cost function for each observation, where the new feature is $\boldsymbol{X}_{n\cdot} - \boldsymbol{Z}_{n\cdot}\boldsymbol{A}$ for observation n. We refer to this algorithm as BpMeans. In light of the K-means++ initialization routine [Arthur and Vassilvitskii, 2007], Broderick *et al.* [2013a] propose an analogous initialization routine that we refer to as BpMeans++ and describe in Algorithm 5. K-means++

Algorithm 5 The BpMeans++ initialization algorithm.

1. start by assigning every observation to the first feature, and let the first feature be the mean of the data
2. recursively, for feature $k > 1$, calculate the distance from each observation $\boldsymbol{X}_{n\cdot}$ to its feature representation $\boldsymbol{Z}_{n\cdot}\boldsymbol{A}$ for the current $\boldsymbol{Z}$ and $\boldsymbol{A}$, and choose an observation $\boldsymbol{X}_{i\cdot}$ with probability proportional to this distance squared
3. assign $\boldsymbol{A}_{k\cdot}$ to be the chosen observation $\boldsymbol{X}_{i\cdot}$
4. assign z_{mk} for all $m \in [N]$ to optimize Eq. 4.27

specifies a similar initialization routine for the K-means algorithm and guarantees that the final K-means objective function will be within a constant factor of the optimal value. The BpMeans++ algorithm, while an empirically beneficial contribution, currently does not imply any optimality guarantees. Determining whether an initialization routine can provide an optimality guarantee for the BpMeans objective is currently an open question. The per-iteration complexity of BpMeans is dominated by checking whether each z_{nk} value improves the objective function, which can be formulated as a $O(KD)$ operation for each n, k pair, yielding a per-iteration complexity of $O(NK_+^2 D)$.

There are several IBP inference techniques that we do not consider here:

- Doshi-Velez *et al.* [2009a] proposed a parallelized IBP sampler that operates via an approximate, loopy message passing scheme. We do not include this inference method because (1) it is a different flavor than all other inference techniques mentioned here as it uses distributed computation and (2) it is very difficult to implement correctly, and the authors have not made their experimental implementation publicly available.

Table 4.2: Worst-case per-iteration complexity given a linear-Gaussian likelihood model for N D-dimensional observations, K_+ active latent features, and parameter ϵ.

Algorithm	Iteration Complexity
MEIBP	$O(NK_+^2(D + \frac{1}{\epsilon}\log(K_+)))$
VIBP [Doshi-Velez *et al.*, 2009b]	$O(NK_+^2 D)$
AIBP [Doshi-Velez and Ghahramani, 2009]	$O(N(K_+^2 + K_+ D))$
BpMeans [Broderick *et al.*, 2013a]	$O(NK_+^2 D)$
UGibbs [Doshi-Velez and Ghahramani, 2009]	$O(NK_+^2 D)$
BS-IBP [Rai and Daume III, 2011]	$O(N^3(K_+ + D)2^{K_+})$
INMF [Ding *et al.*, 2010]	$O(N(K_+^3 D + K_+ D^2))$

- Wood and Griffiths [2007] proposed a particle sampler based inference technique that can be applied to non-conjugate IBP models. However, a number of papers have compared against this inference technique and found that it performs erratically [Doshi-Velez *et al.*, 2009b; Doshi-Velez, 2009; Rai and Daume III, 2011].

- Griffiths and Ghahramani [2006] originally proposed a collapsed IBP sampler in which the latent factors $\boldsymbol{A}$ are analytically integrated out of the model and inference is performed solely on $\boldsymbol{Z}$. This inference method scales cubically in N and is not computationally feasible for $N > 1000$ [Doshi-Velez *et al.*, 2009b].

4.9 Inference Experiments

We evaluated the inference quality and efficiency of MEIBP inference on two synthetic and three real-world datasets. We used the predictive likelihood estimates and L_2 error of held-out observations as our performance criteria and compared MEIBP inference with the methods listed in Table 4.3 (the finite and infinite VIBP are differentiated with an "f-" and "i-" prefix). A discussion of the predictive likelihood estimates can be found in Appendix A.3.5. We used a truncated Gaussian prior on the latent factors for UGibbs and INMF, and Gaussian priors for the AIBP, BpMeans, and variational methods. In our evaluations, we also

included Schmidt *et al.* [2009]'s iterated conditional modes algorithm, which computes a MAP estimate of a parametric nonnegative matrix factorization model: $\boldsymbol{X} = \mathbf{B}\boldsymbol{A} + \mathbf{E}$, where $\mathbf{B}$ and $\boldsymbol{A}$ have exponential priors and $\mathbf{E}$ is zero-mean Gaussian noise. We abbreviate this model "BNMF"; it has a per-iteration complexity of $O(N(K_+^2 + K_+ D))$, where K_+ is the exact number of latent features, not an upper bound. We also used MEIBP as an initialization routine for the AIBP, and denote this inference method as "ME-AIBP."

The VIBP and MEIBP inference methods specify a maximum K value, while the sampling methods and BpMeans are unbounded.[1] Therefore, we also included truncated versions of the sampling methods (indicated by a "t-" prefix) for a fairer comparison and used the bounded BpMeans variant, where the number of latent features cannot exceed a specified bound. We centered all input data to have a 0-mean and unit variance for the models with 0-mean Gaussian priors and a 0-minimum and unit variance for nonnegative models. All inferred matrices were initialized randomly from their respective priors. Following Doshi-Velez and Ghahramani [2009], we fixed the hyperparameters $\sigma_{\boldsymbol{X}}$ and $\sigma_{\boldsymbol{A}}$ to $\frac{3}{4}\sigma$, where σ was the standard deviation across all dimensions of the data, and set $\alpha = 3$. We ran each algorithm until the multiplicative difference of the average training log-likelihood differed by less than 10^{-4} between blocks of five iterations with a maximum runtime of 36 hours. Our experiments used optimized MATLAB implementations of the algorithms, as provided by the respective authors,[2] on 3.20 GHz processors.

4.9.1 Synthetic Data Experiments: Predictive Likelihood and L_2 Error

We created high-noise synthetic datasets in the following way: (1) sample $z_{n,k} \sim \text{Bernoulli}(p = 0.4)$, (2) generate $\boldsymbol{A}$ with K random, potentially overlapping binary factors, (3) let $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A} + \mathbf{E}$, where $\mathbf{E} \sim \mathcal{N}(0, 1)$. We evaluated the predictive likelihood and L_2 error on a held-out portion of 20% of the dimensions from the

[1] NB: the sampling methods sample from the true unbounded posterior while the BpMeans uses a heuristic rule to add new features.

[2] Except the BpMeans algorithm, which the authors did not publicly release, and we therefore wrote the inference procedure ourselves.

Table 4.3: Summary of inference algorithms used in these experiments. The "Factor prior" column shows the prior distribution specified for the latent factors, where "G" is Gaussian, "TG" is truncated Gaussian, and "Exp" is exponential.

Algorithm	Reference	Description	Factor prior	Iteration complexity
meibp	this work	maximization expectation IBP	TG	$O(NK_+^2(D + \frac{1}{\epsilon}\log(K_+)))$
me-aibp	this work	maximization expectation IBP initialization for AIBP	TG + G	$O(NK_+^2(D + \frac{1}{\epsilon}\log(K_+)))$
i-vibp	Doshi-Velez *et al.* [2009b]	unbounded variational IBP	G	$O(NK_+^2D)$
f-vibp	Doshi-Velez *et al.* [2009b]	finite variational IBP	G	$O(NK_+^2D)$
aibp	Doshi-Velez and Ghahramani [2009]	unbounded accelerated gibbs sampling	G	$O(N(K_+^2 + K_+D))$
t-aibp	Doshi-Velez and Ghahramani [2009]	truncated accelerated gibbs sampling	G	$O(N(K_+^2 + K_+D))$
bpmeans	Broderick *et al.* [2013a]	0-variance MAP estimate	G	$O(NK_+^2D)$
ugibbs	Doshi-Velez and Ghahramani [2009]	unbounded uncollapsed Gibbs sampling	G	$O(NK_+^2D)$
t-ugibbs	Doshi-Velez and Ghahramani [2009]	truncated uncollapsed Gibbs sampling	TG	$O(NK_+^2D)$
bnmf	Schmidt *et al.* [2009]	parametric matrix factorization with exponential priors	Exp	$O(N(K_+^2 + K_+D))$

last half of the data.

Figure 4.10 shows the change of the test log-likelihood and L_2 error over time for a small dataset with $N = 500, D = 500, K = 20$ (2.5×10^5 total observations), while Figure 4.11 shows a similar plot for a much larger dataset with $N = 10^5, D = 10^3, K = 50$ (10^8 total observations). We initialized all models with the true number of latent features. The error regions display the standard deviation over five random restarts.[1] The BS-IBP and INMF methods were removed

[1]Over half of the tests from the VIBP methods converged after one iteration to a very poor local optimum. We did not include these outcomes in any of our experiments.

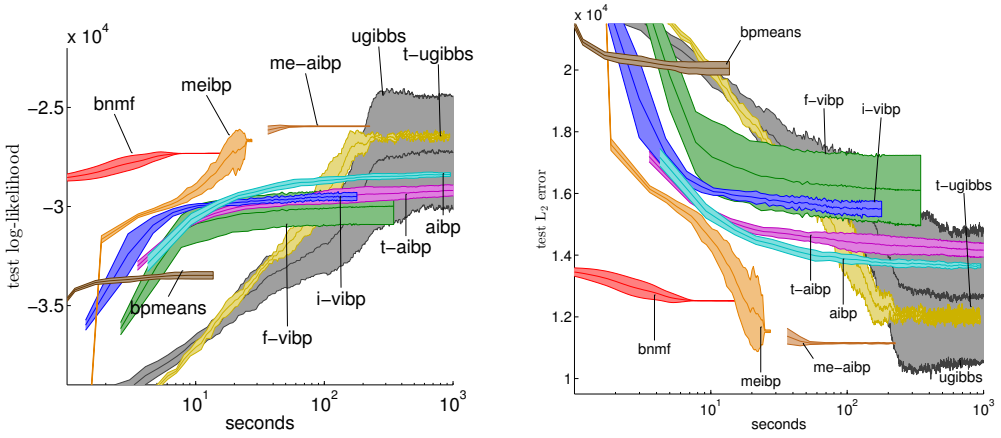


Figure 4.10: Evolution of test log-likelihood (left) and L_2 error (right) for a synthetic dataset of size $N = 500,\ D = 500,\ K = 20$.

from our experiments following the synthetic dataset tests as both methods took at least an order of magnitude longer than the other methods: in 36 hours, the BS-IBP did not complete a single iteration on the small dataset, and the INMF did not complete a single iteration on the large dataset.

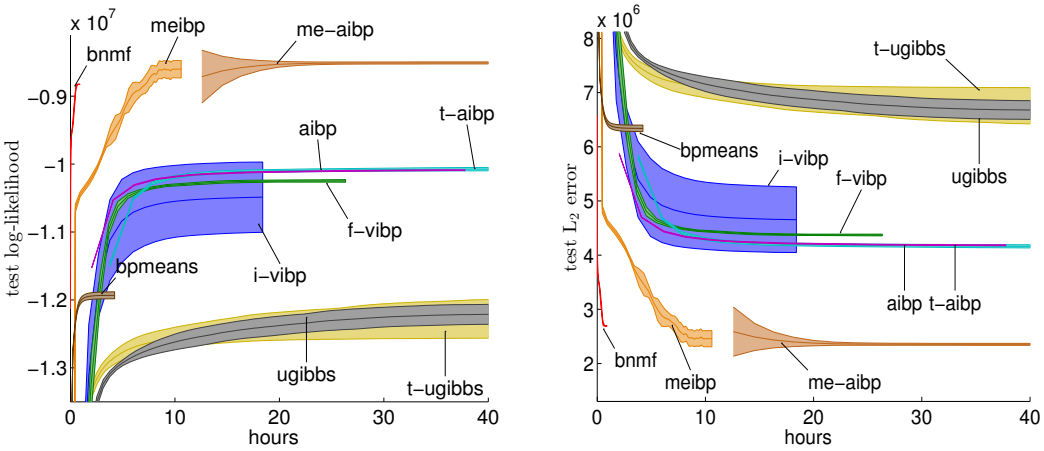


Figure 4.11: Evolution of test log-likelihood (left) and L_2 error (right) for a synthetic dataset of size $N = 10^5,\ D = 10^3,\ K = 50$.

BpMeans converged quickest among the IBP models, however its maximum likelihood approach caused it to get stuck in a relatively poor local optima for both the small and large datasets. The MEIBP, on the other hand, converged

to the same nearly the same test-likelihood as the uncollapsed samplers, but an order of magnitude faster. Its advantage over the BpMeans in this situation stemmed from its ability to maintain uncertainty about the model parameters through its variational distributions, i.e. it is not a fully MAP approach and was less proned to local optima. BpMeans remained virtually unchanged after its BpMeans++ initialization (Algorithm 5), while the MEIBP progressed to better solutions following its random initialization. The MEIBP worked well as an intialization for AIBP: the mean predictive likelihood and L_2 error of ME-AIBP were best among all inference techniques for the small and large datasets. The AIBP was unable to obtain the same performance from a random initialization.

Though BNMF specified exponential priors on the factorized matrices, it was able to perform relatively well by inferring bimodal matrices similar to the generated data matrices, see Figure 4.12. Since the exponential priors were not binary, however, linearly combining the factorized matrices resulted in a stronger averaging effect than the IBP models. As shown in Figure 4.13, the BNMF assigned average values to more observations in comparison to MEIBP, which also held true for UGibbs, and resulted in a Gaussian-like distribution of values with a smaller variance than the IBP models. For the small synthetic dataset, this averaging caused BNMF to perform worse than MEIBP and UGibbs. For certain datasets, however, the flexibility in the prior that leads to this type of averaging may be beneficial.

For the small synthetic dataset, the VIBP methods converged quicker than the samplers but had trouble escaping local optima. Doshi-Velez *et al.* [2009b] noted a similar result: the variational techniques struggled on smaller datasets, while the sampling techniques tended to perform well. The authors also showed that the variational techniques tended to outperform the sampling techniques on larger data, though they did not compare against AIBP. Our experiments agree with both of these observations, and we add that the AIBP outperformed the variational methods on the synthetic dataset with 10^8 total observations. For the large synthetic dataset, BNMF converged quicker than the IBP models, but this time MEIBP was the only IBP model to outperform BNMF in the given time limit. The UGibbs methods and variational techniques got stuck in relatively poor local optima.

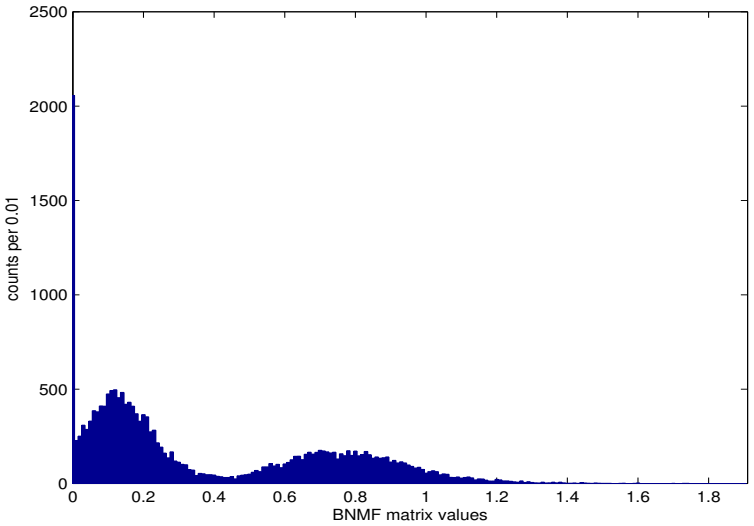


Figure 4.12: Histogram of values from the BNMF factorized matrices from the $N = 500, D = 500, K = 20$ simulated data experiment. Note the bimodal distribution that weakly emulates a Bernoulli (the true factorized matrix distributions).

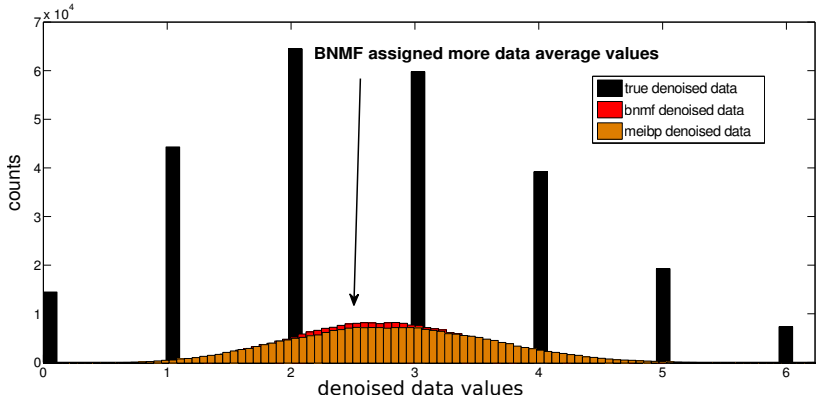


Figure 4.13: Comparison of the inferred denoised values from the $N = 500, D = 500, K = 20$ simulated data experiment. Note the BNMF displays a Gaussian-like distribution of values with a smaller variance in comparison with the MEIBP distribution.

These synthetic data experiments showed that the MEIBP inference performed comparably or better than other MAP, sampling, and variational inference techniques on inferring noisy, linearly-combined factors. Furthermore, if posterior samples are desired, the MEIBP can be used as a structured initialization procedure for samplers. Our real data experiments in the next subsection further support these conclusions. We were surprised by the BNMF performance on these synthetic datasets, and due to its simplicity of implementation and use, practitioners that do not require a "latent features" interpretation of their model should consider this inference technique for matrix factorization.[1]

4.9.2 Real Data Experiments: Predictive Likelihood and L_2 Error

Table 4.4 summarizes the real-world datasets used in our experiments. The Piano and Faces datasets are dense real-valued datasets, whereas the Flickr dataset is a sparse binary dataset (0.81% filled). The Piano and Flickr dataset were previously used for real data experiments by Doshi-Velez and Ghahramani [2009] and Doshi-Velez *et al.* [2009a], while the Faces dataset is similar to the original Yale faces dataset explored by Doshi-Velez and Ghahramani [2009] and Doshi-Velez *et al.* [2009b], but it provided more testing data than the original dataset. For the Piano and Flickr datasets, we evaluated the predictive likelihood on a held-out portion of 20% of the dimensions from the last half of the datasets.[2] The Faces dataset had roughly sixty-four facial images of thirty-eight subjects, and we removed the bottom half of five images from each subject for testing.

Figure 4.14 shows the test log-likelihood and L_2 error evolution on the Piano dataset for all inference models/methods for initialization $K = \{10, 25, 50\}$ (except the uncollapsed IBP sampling methods, which were initialized randomly from the IBP prior with $\alpha = 3$ and are shown in the $K = 50$ plot). The error bars indicate the standard deviation in performance over the five best random restarts from eight trials.

[1] At the time of this writing, a MATLAB BNMF implementation is available from the author at http://mikkelschmidt.dk/uploads/media/bayesnmf.zip.

[2] For the Flickr data, we required each column to have at least five non-zero entries: this requirement removed 92 dimensions.

Table 4.4: Summary of real-world datasets.

Dataset	Size ($N \times D$)	Details
Piano [Poliner and Ellis, 2006]	16000×161	DFT magnitudes of piano recordings
Faces [Lee *et al.*, 2005]	2414×32256	face images with various lightings
Flickr [Kollar and Roy, 2009]	25000×1500	binary image-tag indicators

Overall, the Piano results were similar to the small synthetic dataset. The BNMF converged much faster than the IBP models to very good solutions, and the MEIBP and BpMeans performed best among the IBP models in terms of runtime, test log-likelihood, and L_2 error. The strong performance of MEIBP and BNMF likely stems from the inherent nonnegativity of the DFT magnitudes, and it appears that the Piano dataset was amenable to MAP solutions as demonstrated by the strong performance of all MAP techniques (BNMF, BpMeans, and MEIBP).

We noticed interesting behavior in the $K = 50$ plots: the unbounded AIBP sampler eventually performed comparable to the MAP approaches, but it took an order of magnitude longer to obtain similar results. For the $K = 50$ experiments, the BNMF quickly obtained very good results, but as inference continued, it overfit to the training data because it continued to optimize all $K = 50$ features to the training data (it is a parametric model). The MEIBP, on the other hand, began with $K = 50$, and for some runs, it removed unneeded features and finished with circa $K = 45$ features.

Similar to the synthetic dataset tests, the ME-AIBP technique obtained the best predictive likelihood and L_2 performance among all tested methods. This technique attests to the strength of using a structured initialization routine for sampling techniques. The MEIBP was itself initialized randomly, but it may be beneficial to initialize it using the BpMeans++ initialization routine. We leave this question for future research.

The uncollapsed Gibbs techniques performed poorest among all methods. The Piano dataset, and indeed all datasets in this subsection, were too large for uncollapsed Gibbs sampling to be effective. The uncollapsed sampler draws feature assignments from a massive combinatorial state space, and then conditioned on

these features, it sampled factors from a massive real-valued state space. When initialized randomly, the uncollapsed sampler was consistently unable to escape poor local optima. We experimented with initializing the uncollapsed sampler with MEIBP and obtained similar result to ME-AIBP. We did not formalize these experiments, however, so we only mention it as a possibly useful inference technique for large combinatorial models.

Figure 4.16 shows the inference results for the Faces dataset for the $K = \{10, 25, 50\}$ initialization. BNMF is not present on any of these plots as its predictive likelihood results hovered around -6×10^6. This resulted because the update steps in the BNMF technique updated the factorized matrices for each of the dimensions of the data, and removing a sequence of half of the dimensions from the images caused BNMF to simply return mean estimates of the training data for the test data predictions. We tried reordering the BNMF updates to avoid this result but did not have success. BpMeans is not present for the $K = 50$ plot as it experienced overfitting problems with higher K bounds and produced predictive-likelihood results that hovered around -4.7×10^6.

With faces from 38 different humans under various lighting conditions, all IBP inference techniques inferred features that focused on lighting conditions more than facial expressions or facial features. This resulted for a few reasons: (1) the lighting effects caused greater variation in the pixel values of the images compared to the facial expressions and features (2) the Faces dataset contained only one facial pose (front-center) for each of the thirty-eight subjects under sixty-four different lighting conditions, since the lighting conditions were the same for all subjects, they provided a parsimonious representation of the data. Figure 4.15 shows typical inference results for MEIBP, and the sampling/variational results looked very similar, i.e. the features emphasized the lighting effects instead of the facial features.

$K = 10$ was too small of a bound to capture the various lightings in the faces dataset. All inference techniques performed relatively poorly with this bound and inferred vague left, right, top, or bottom shadow features. The BpMeans++ initialization routine worked well as it initialized the latent factors to actual faces with different lighting effects. The inference procedure then softened the factors so they would better fit the observed data. With $K = 50$, however, the number

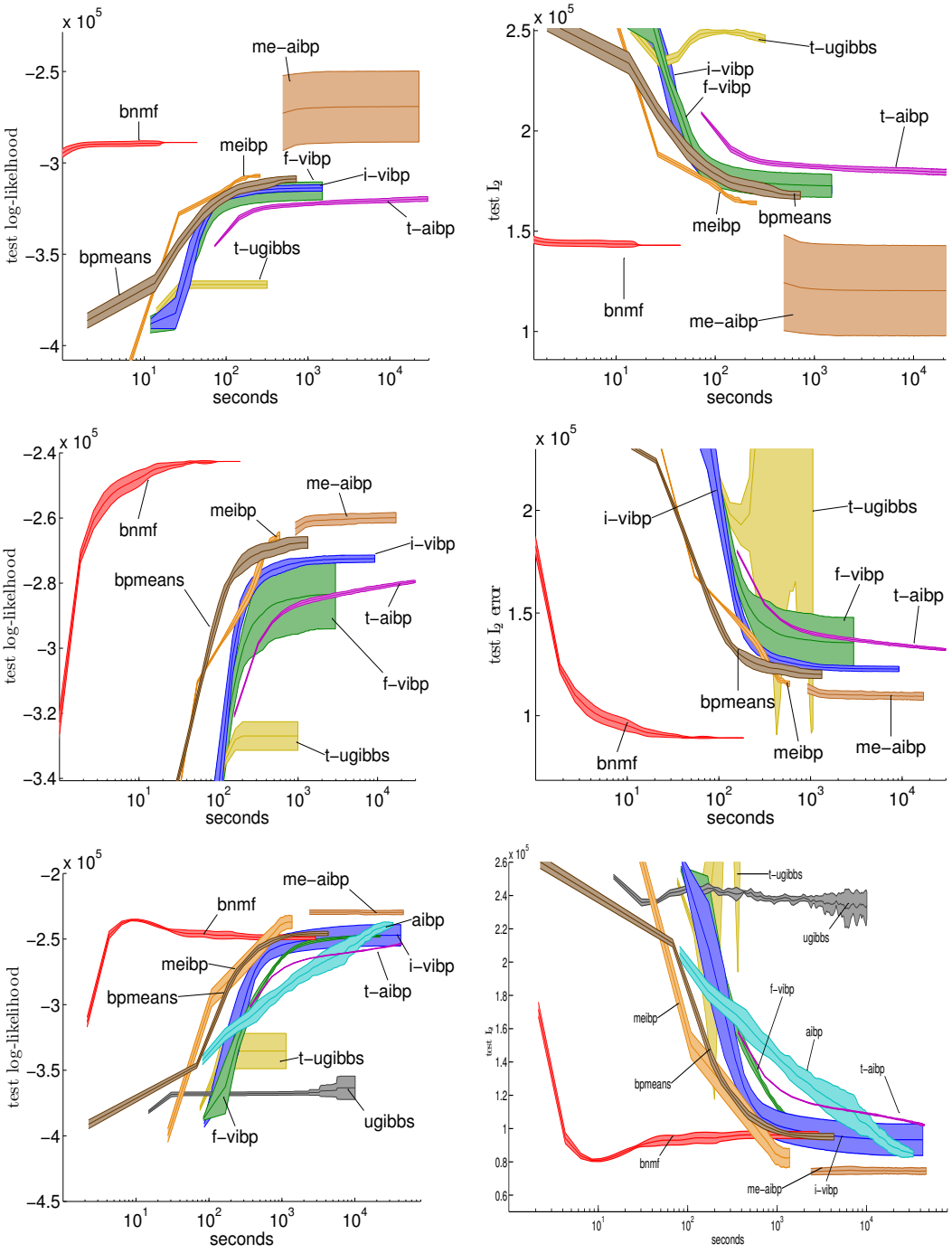


Figure 4.14: Evolution of test log-likelihood (left) and L_2 error (right) for the piano dataset. The top row has $K = 10$ initialization, the middle row has $K = 25$ initialization, and the bottom row has $K = 50$ initialization. The $K = 50$ initialization plots also include the unbounded samplers.

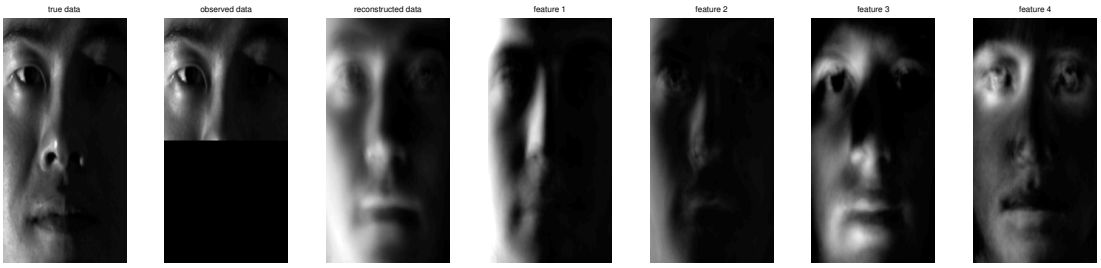


Figure 4.15: Example Faces inference result for the MEIBP. From left to right: True face observation from the Faces data, input data (bottom half of the image was masked), reconstructed data, four latent factors.

of factors exceeded the distinct number of lighting effects and the model began at a poor optimum, where factors were overfit to specific data.

With $K = 25$ and $K = 50$, the MEIBP technique struggled to improve upon its first round of optimizations. By iteratively inspecting the MEIBP factors and feature assignments throughout its inference, we believe these results were due to the nonnegative prior on the latent factors. After its first optimization round, the MEIBP initially expressed the data via nonnegative shadow factors, and in subsequent optimization rounds it added or removed features to improve its shadow representations of the training data and updated its factors accordingly. However, since it could not create negative factors, adding/removing features from an observation added/removed mass from the associated factors, which inversely affected other observations. The Gaussian priors for the variational and accelerated Gibbs methods allowed the observations to add features that removed mass from the associated factors, and in turn, this gave them more flexibility to overcome local optima in comparison to the nonnegative factors.

Figure 4.17 shows the inference result on the binary Flickr dataset: a metadata dataset of image label tags. The Flickr dataset was sparse (0.81% filled), and the inference challenge was to infer non-zero values for the test data. Like the previous datasets, the Flickr dataset was normalized to unit variance and zero mean for the models with Gaussian priors on the latent factors and unit variance and zero minimum for models with nonnegative factor priors. In this case, this normalization led to observed image tags taking values that were greater than one and unobserved tags taking values that were either zero or negative. All

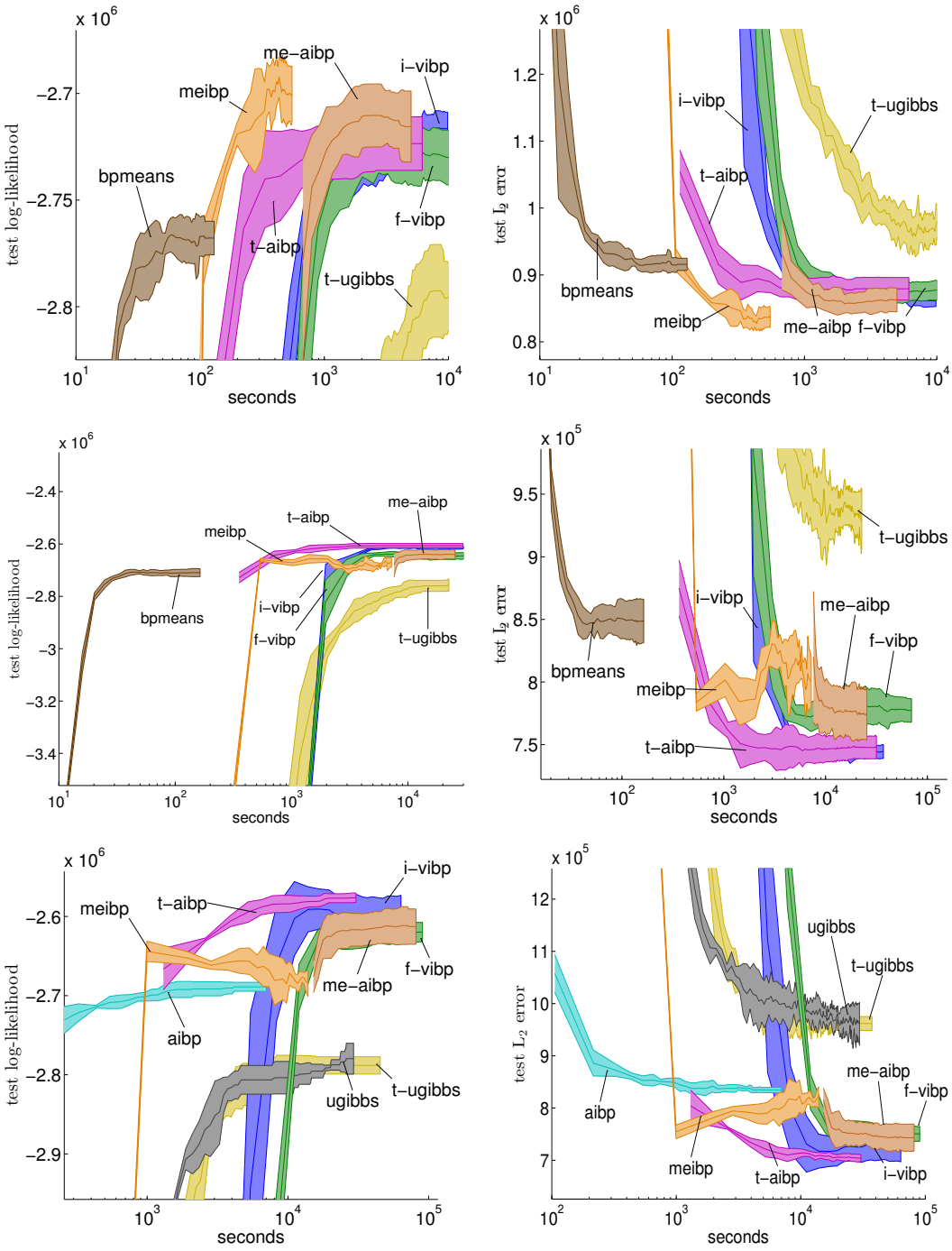


Figure 4.16: Evolution of test log-likelihood (left) and L_2 error (right) for the Face dataset. The top row has $K = 10$ initialization, the middle row has $K = 25$ initialization, and the bottom row has $K = 50$ initialization. The $K = 50$ initialization plots also include the unbounded samplers.

inference techniques demonstrated greater variance in the test likelihood and L_2 error for the Flickr dataset, indicating that the initialization played a substantial role in the final inference results. For reference, a trivial baseline of using 0 for all test data yields an L_2 error of 3.157×10^6.

For all K initialization values, the IBP inference methods typically had one active feature for 1000-2000 observations and the rest of the features were active for less than fifty observations. This occurred because the observations were binary valued and one latent feature could express the majority of the data as the corresponding latent factor equalled the magnitude of the observed data. The remaining latent features were typically active when the main latent feature was not active and the associated factor magnitudes were smaller than the factor values corresponding to the main factor. In other words, the less popular latent features were used to capture the boundary-case image tags, where a given observation could activate a number of smaller factors to reflect its belief that a given image tag was active. By increasing the K bound, all IBP methods had a greater resolution to describe the inferred belief of a given image tag being active, and as a result, all methods (modulo the uncollapsed Gibbs sampler) performed better as K increased.

For the ME-AIBP method, the initial Gibbs sweep changed over half of the feature assignments for $K = 10$, which resulted in a substantial degradation in predictive likelihood and L_2 error. For $K = 50$, the initial Gibbs sweep changed less than five percent of the latent feature assignments, and consequently, the ME-AIBP inference method obtained the best predictive-likelihood and L_2 error across all K values and inference techniques. As in this case, examining the agreement between the final MEIBP and initial sampling results may indicate whether the MEIBP provides a useful initialization.

UGibbs performed erratically in L_2 error yet consistently (but poorly) in predictive likelihood. This occured because UGibbs obtained at least one reasonable sample within each block of ten samples. The test likelihood estimated by averaging these samples (see Appendix A.3.5) was consistent because the single good sample dominated the average. The UGibbs predictive likelihood then remained consistent when averaged over the five randomly-restarted trials. At any given sampling step, however, one of the five UGibbs runs had a very poor sample that

dominated the average and led to poor L_2 performance. We note that a Gaussian likelihood is not ideal for binary data, and we expect that specifying a model that uses one of the binary likelihoods mentioned in the next chapter would help improve the inference results on the Flickr dataset.

In the above experiments, the MEIBP often exhibited a sudden convergence whereby it obtained a local optimum, and the ls-algorithm did not change any $\boldsymbol{Z}$ assignments. This is a characteristic of using hard assignments with a greedy algorithm: at a certain point, changing any latent feature assignments decreased the objective function. This abrupt convergence, in combination with the speed of the submodular maximization algorithm, helped the MEIBP consistently converge faster than the sampling and variational IBP methods. Furthermore, the submodular maximization algorithm converged to local optima that were comparable or better than the sampling or variational results, though at the cost of only obtaining a MAP solution. Like the variational methods, it maintained a distribution over $\boldsymbol{A}$ that prevented it from getting stuck in local optima early on, and like the sampling methods, the MEIBP used hard $\boldsymbol{Z}$ assignments to take larger steps in the inference space and obtain better optima for some datasets. Of course, the MEIBP is not the holy grail, but we view it as evidence that submodular optimization can be exploited for fast inference in latent feature models that obtains solutions comparable to samplers or variational techniques.

4.9.3 Finding the True Number of Latent Features

An ostensible advantage of using nonparametric priors is that a user does not need to specify the multiplicity of the prior parameters. Rather, the user specifies a stochastic process that serves as a prior on an unbounded parameter space. Clever sampling techniques such as slice sampling and retrospective sampling allow samples to be drawn from these nonparametric priors, c.f. Teh *et al.* [2007] and Papaspiliopoulos and Roberts [2008].

As mentioned previously, variational methods are not directly amenable to Bayesian nonparametric priors as the variational optimization cannot be performed over an unbounded prior space. Instead, variational methods must specify a maximum model complexity (parameter multiplicity). Several heuristics

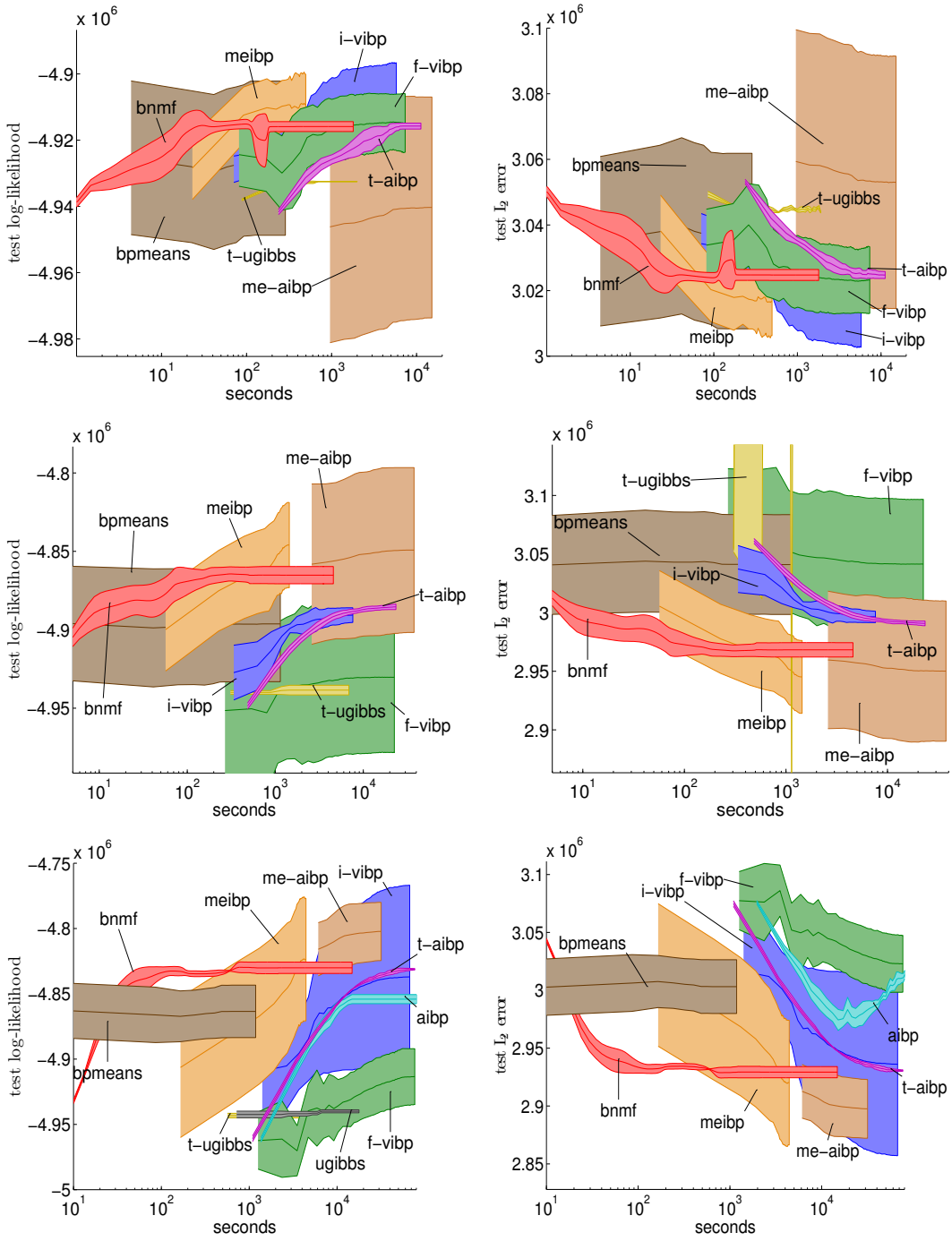


Figure 4.17: Evolution of test log-likelihood (left) and L_2 error (right) for the Flickr dataset. The top row has $K = 10$ initialization, the middle row has $K = 25$ initialization, and the bottom row has $K = 50$ initialization. The $K = 50$ initialization plots also include the unbounded samplers.

have been proposed to address this limitation: Wang and Blei [2012a] sampled from the variational distribution for the local parameters—which included sampling from the unbounded prior—and used the empirical distributions of the local samples to update the global parameters, while Ding *et al.* [2010] simply started with $K_+ = 1$ and greedily added features. We did not address these techniques in this work as the MEIBP performed competitively with the unbounded sampling techniques without employing these types of heuristics. Furthermore, here we demonstrate that the MEIBP can robustly infer the true number of latent features when the K_+ bound is greater than the true number of latent features.

For this experiment, we generated the binary images dataset used in Griffiths and Ghahramani [2005], where the dataset, $\boldsymbol{X}$, consisted of 2000 6×6 images. Each row of $\boldsymbol{X}$ was a 36 dimensional vector of pixel intensity values that was generated by using $\boldsymbol{Z}$ to linearly combine a subset of the four binary factors shown in Figure 4.18. Gaussian white noise, $\mathcal{N}(0, \sigma_X)$, was then added to each image, yielding $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{A} + \mathbf{E}$. The feature vectors, $\boldsymbol{Z}_{n\cdot}$ were sampled from a distribution in which each factor was present with probability 0.5. Figure 4.19 shows four of these images with different σ_X values.

Figure 4.18: The four binary latent factors used in the sensitivity analysis in this section. The white squares are ones and the dark squares are zeros.

We initialized the MEIBP with $K = 20$, σ_X=1.0, $\sigma_A = 1.0$, $\alpha = 2$, $\tilde{\mu}_{kd} \sim \mathcal{TN}(0, 0.05)$ (variational factor means), $\tilde{\sigma}_{kd} \sim \mathcal{TN}(0, 0.1)$ (variational factor standard deviations), $z_{nk} \sim \text{Bernoulli}(\frac{1}{3})$. With this initialization, we tested the MEIBP robustness by performing MEIBP inference on $\boldsymbol{X}$ for $\sigma_X = 0.1, \ldots, 1.0$ in 100 evenly spaced increments with all hyperparameters and algorithm options unchanged during the experiment. MEIBP convergence was determined in the same way as in the previous experimental section, i.e. when the mean of the test likelihood between blocks of five iterations differed by a relative factor of less than 10^{-4}. Figure 4.20 (left) shows a histogram of the final number of MEIBP features ($K_{\text{true}} = 4$) and Figure 4.20 (right) shows the final number of MEIBP

features as a function of σ_X.

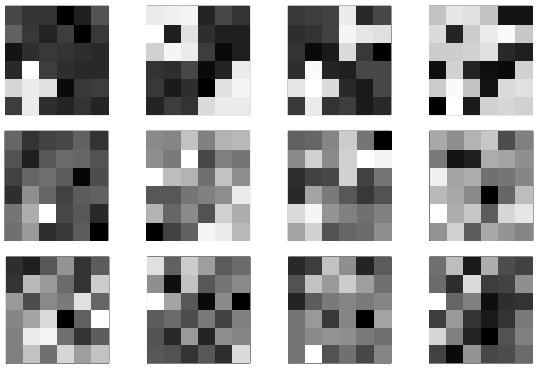


Figure 4.19: Example data used in the sensitivity analysis. Each column contains the same combination of latent factors, where the top row has a data noise term of $\sigma_X = 0.1$, the middle row has $\sigma_X = 0.5$, and the bottom row has $\sigma_X = 1.0$.

These results indicate that the regularizing nature of the IBP prior tends to lead to the correct number of latent features even when the K_+ bound is much larger than the true K_+. Furthermore this experiment indicates that MEIBP inference is robust to model noise, at least, for the simple data used in this experiment. At a medium level of data noise, the inference occasionally finished with $K_+ = 3$, which resulted from two true latent factors collapsing to the same inferred latent feature. Once this occurred, MEIBP did not have a mechanism for splitting the features. For σ_X comparable to the latent factors, $\sigma_X \geq 0.9$, MEIBP often inferred "noise features," which were essentially white noise. These features were typically active for less than 4% of the data instances.

4.10 Chapter Summary

In this chapter we used the IBP submodularity results of Chapter 3 in combination with Kurihara and Welling [2008]'s ME framework to perform approximate MAP inference via a sequence of submodular maximizations for each observation's feature assignments. Our key insight was to exploit the submodularity inherent in the evidence lower bound formulated in §4.2, which arose from the quadratic pseudo-Boolean component of the linear-Gaussian model. We explored various

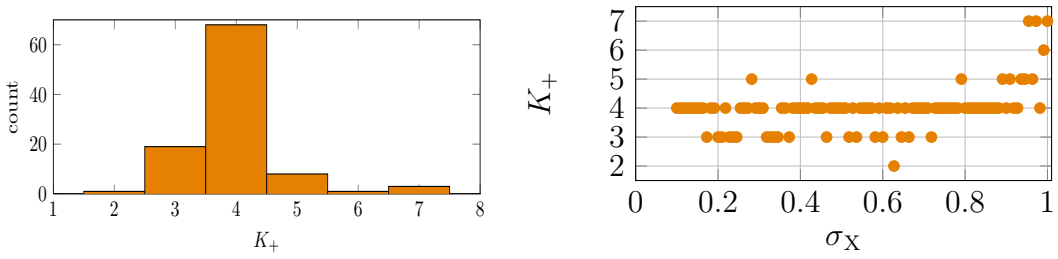


Figure 4.20: Final feature count (K_+ value) for MEIBP inference where $K_{\text{true}} = 4$ for the binary image data with K_+ initialized to 20 for $\sigma_X = 0.1, \ldots, 1.0$ in 100 evenly spaced increments with all hyperparameters and algorithm options fixed during the experiment. (**Left**) histogram of final K_+ values. (**Right**) final K_+ values as a function of σ_X.

submodular optimization algorithms and chose the linear search method from Feige *et al.* [2011] as it performed best on our controlled experiments. MEIBP inference converged faster than competing IBP methods and obtained comparable solutions on various datasets. A simple, documented, and supported MATLAB implementation of MEIBP is available at `https://github.com/cjrd/MEIBP`, and the less-simple, undocumented experimental code used in this section is available at `http://mlg.eng.cam.ac.uk/~creed/`.

Chapter 5

Log-Submodular Latent Feature Models

In this chapter we outline a number of models that can benefit from the submodularity results of this thesis. Specifically, we show certain forms of the following types of models are log-submodular for each observation's feature assignments:

- Sparse matrix factorization models
- Latent attribute models for network data
- Leaky, noisy-or model for binary data

We do not specify an exact inference framework for these models as we did with the linear-Gaussian model in the previous chapter. Rather, we show the joint-probability of these models is log-submodular for each observation's feature assignments and discuss various inference frameworks in §5.4.

5.1 Sparse Matrix Factorization Models

We can generalize the IBP linear-Gaussian model studied in the previous chapter by allowing the factor loadings to take on nonnegative weights and specifying an arbitrary nonnegative prior on the factors. This class of sparse matrix factoriza-

tion models has the following generative process:

$$\boldsymbol{Z}|\alpha,\beta \sim \text{IBP}(\alpha,\beta) \tag{5.1}$$
$$\boldsymbol{W}|\theta_1 \sim H_1(\theta_1) \tag{5.2}$$
$$\boldsymbol{A}|\theta_2 \sim H_2(\theta_2) \tag{5.3}$$
$$\boldsymbol{X}_{n\cdot}|\sigma_{\boldsymbol{X}} \sim \mathcal{N}\left((\boldsymbol{W}_{n\cdot} \odot \boldsymbol{Z}_{n\cdot})\boldsymbol{A}, \sigma_{\boldsymbol{X}} I\right), \tag{5.4}$$

where $H_i, i = \{1,2\}$ are nonnegative distributions with parameters $\theta_i, i = \{1,2\}$, $\boldsymbol{W}$ is an $N \times K$ nonnegative weight matrix, and $\odot$ represents the element-wise product. This class of sparse matrix factorization models yields "infinite" independent component analysis and factor analysis models by selecting the appropriate nonnegative priors on the factors and factor loadings [Knowles and Ghahramani, 2007]. The joint probability of this model factorizes as

$$p(\boldsymbol{Z},\boldsymbol{X},\boldsymbol{W}\boldsymbol{A}|\boldsymbol{\theta}) = p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{W},\boldsymbol{\theta})p([\boldsymbol{Z}]|\boldsymbol{\theta})p(\boldsymbol{A}|\boldsymbol{\theta})p(\boldsymbol{W}|\boldsymbol{\theta}), \tag{5.5}$$

where $\boldsymbol{\theta}$ represents the model parameters. Since we know that $\log\left(p([\boldsymbol{Z}]|\boldsymbol{\theta})\right)$ is a submodular function for each observation's feature assignment and $\boldsymbol{W}$ and $\boldsymbol{A}$ only depend on $\boldsymbol{Z}$ through its K_+ selection, all that remains is to show that $p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{W},\boldsymbol{\theta})$ is log submodular for each observation's feature assignments.

$$\log\left(p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{W},\boldsymbol{\theta})\right) = \frac{-1}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}||\boldsymbol{X}_{n\cdot} - (\boldsymbol{W}_{n\cdot} \odot \boldsymbol{Z}_{n\cdot})\boldsymbol{A}|| - \frac{ND}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) \tag{5.6}$$

which is a quadratic pseudo-Boolean function of $\boldsymbol{Z}_{n\cdot}$ with fixed $\boldsymbol{W},\boldsymbol{A}$. From Theorem 1, a quadratic pseudo-Boolean function is submodular if and only if the weight matrix is nonpositive. Since we specified $\boldsymbol{A}$ to be nonnegative, the negative scalar $\frac{-1}{2\sigma_{\boldsymbol{X}}^2}$ and nonnegative weights $\boldsymbol{W}$ ensure that the quadratic component, $(\boldsymbol{Z}_{n\cdot} \odot \boldsymbol{W}_{n\cdot})\boldsymbol{A}\boldsymbol{A}^T(\boldsymbol{Z}_{n\cdot} \odot \boldsymbol{W}_{n\cdot})^T$ is nonpositive. Therefore the joint probability of this class of sparse matrix factorization models is log-submodular for each observation's feature assignments.

5.2 Latent Attribute Model for Network Data

Palla *et al.* [2012] defined an infinite latent attribute model (ILA) for network data that takes the form of unweighted, undirected graphs, where the observed data is a binary $N \times N$ adjacency matrix. ILA assumes each observation in the network has a set of latent features, and if an observation has a specific latent feature, then the object is assigned to a subcluster of that feature. ILA then uses weights between the subclusters of each feature to determine the probability of a link between any two observations.

Palla *et al.* [2012] use the following intuitive example to motivate this model: consider a network of individuals in a university, where a link between two individuals denotes a friendship or acquaintance. Here, the features could indicate whether a student "plays football" or "belongs to a college." The features could then be divided into many different subclusters: "plays football for team X," "belongs to college Y," etc. In ILA, the probability of interaction for each pair of students depends on their subcluster assignments. Formally, the generative model is as follows:

$$\boldsymbol{Z}|\alpha \sim \text{IBP}(\alpha) \tag{5.7}$$

$$\mathbf{c}^{(k)}|\gamma \sim \text{CRP}(\gamma) \tag{5.8}$$

$$w_{st}^{k}|\sigma_w \sim \mathcal{N}(0, \sigma_w) \tag{5.9}$$

$$r_{ij}|\boldsymbol{Z}, \boldsymbol{C}, \boldsymbol{W} \sim \text{Bernoulli}\left(\sigma\left(\sum_{m} z_{im} z_{jm} w^{m}_{c_i^m c_j^m} + s\right)\right) \tag{5.10}$$

where $\boldsymbol{R}$ is the observed $N \times N$ binary adjacency matrix, w_{st}^{k} is the affinity weight between subcluster s and subcluster t within feature k, $\boldsymbol{c}^k$ is the subcluster assignments of all observations with feature k, and $\boldsymbol{Z}$ is a binary $N \times K_+$ feature assignment matrix. This probabilistic model is log-submodular for each observation's feature assignment if we require the subcluster weights to be nonnegative. Nonnegative weights imply that subclusters can only express an affinity for each other (nonnegative subcluster weights), not a repulsion (negative subcluster weight). We use the moniker pILA to refer to the positive-weight variant of ILA. We prove the submodularity of pILA in the remainder of this section.

To show the log-submodularity of pILA we only need to show that the likelihood is a log-submodular function of each observation's feature assignments because the IBP prior is log submodular, and the other probabilistic components only depend on $\boldsymbol{Z}$ through the K_+ selection. The log-likelihood of pILA is

$$\begin{aligned}\log\left(P(\boldsymbol{R}|\boldsymbol{Z},\boldsymbol{C},\boldsymbol{W})\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} -r_{ij}\log\left(1+\exp\left(-\sum_{k=1}^{K_+} z_{ik}z_{jk}w^k_{c_i^k c_j^k} - s\right)\right) \\ +(1-r_{ij})\log\left(1-\frac{1}{1+\exp\left(-\sum_{k=1}^{K_+} z_{ik}z_{jk}w^k_{c_i^k c_j^k} - s\right)}\right),\end{aligned} \tag{5.11}$$

where for a particular $\boldsymbol{Z}_{n\cdot}$, the objective function is

$$\begin{aligned}\mathcal{F}(\boldsymbol{Z}_{n\cdot}) = &\sum_{j=1}^{N} -r_{nj}\log\left(1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_n^k c_j^k} - s\right)\right) \\ &+ (1-r_{nj})\log\left(1-\frac{1}{1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_n^k c_j^k} - s\right)}\right) \\ &\sum_{j\in[N]\setminus n} -r_{jn}\log\left(1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_j^k c_n^k} - s\right)\right) \\ &+ (1-r_{jn})\log\left(1-\frac{1}{1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_j^k c_n^k} - s\right)}\right),\end{aligned} \tag{5.12}$$

where the data and affinity weight indices is the key difference between the first and second summations. Since both summations have the same form and nonnegative combinations of submodular functions are submodular, we only need to show the following function is submodular to prove the submodularity of $\mathcal{F}$:

$$\begin{aligned}\mathcal{H}_j(\boldsymbol{Z}_{n\cdot}) = &- r_{nj}\log\left(1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_n^k c_j^k} - s\right)\right) \\ &+ (1-r_{nj})\log\left(1-\frac{1}{1+\exp\left(-\sum_{k=1}^{K_+} z_{nk}z_{jk}w^k_{c_n^k c_j^k} - s\right)}\right)\end{aligned} \tag{5.13}$$

for some arbitrary but specific index j. For set $A_n = \{i : z_{ni} = 1\}$, we can define the equivalent set function:

$$\mathcal{H}_j(\boldsymbol{Z}_{n\cdot}) = \mathcal{G}_j(A_n) = -\, r_{nj}\log\left(1 + \exp\left(-\sum_{k\in A_n} z_{jk} w^k_{c^k_n c^k_j} - s\right)\right) + (1 - r_{nj})\log\left(1 - \frac{1}{1 + \exp\left(-\sum_{k\in A_n} z_{jk} w^k_{c^k_n c^k_j} - s\right)}\right). \tag{5.14}$$

We use the definition of submodularity to show $\mathcal{G}(A_n)$ is submodular by examining cases $r_{nj} = 0$ and $r_{nj} = 1$. We use the following convenience function for the below proof:

$$d_j(A_n) = \sum_{k\in A_n} z_{jk} w^k_{c^k_n c^k_j} + s \tag{5.15}$$

For $r_{nj} = 0$ the definition of submodularity yields the following inequality for sets $A_n \subseteq B_n \subseteq [K_+]$ and element $e \in [K_+] \setminus B_n$:

$$\log\left(\frac{1 - (1 + \exp(-d_j(A_n \cup \{e\})))^{-1}}{1 - (1 + \exp(-d_j(A_n)))^{-1}}\right) \geq \log\left(\frac{1 - (1 + \exp(-d_j(B_n \cup \{e\})))^{-1}}{1 - (1 + \exp(-d_j(B_n)))^{-1}}\right)$$

$$\log\left(\frac{\exp(d_j(A_n)) + 1}{\exp(d_j(A_n \cup \{e\})) + 1}\right) \geq \log\left(\frac{\exp(d_j(B_n)) + 1}{\exp(d_j(B_n \cup \{e\})) + 1}\right)$$

$$(\exp(d_j(B_n \cup \{e\})) + 1)(\exp(d_j(A_n)) + 1) \geq (\exp(d_j(B_n)) + 1)(\exp(d_j(A_n \cup \{e\})) + 1)$$

By expanding $d_j(A_n \cup \{e\})$ to $d_j(A_n) + d_j(\{e\}) - s$ and likewise for B_n we have

$$\exp(d_j(B_n) + d_j(\{e\})) + \exp(d_j(A_n)) \geq \exp(d_j(A_n) + d_j(\{e\})) + \exp(d_j(B_n))$$

$$\exp(d_j(B_n))(\exp(d_j(\{e\})) - 1) \geq \exp(d_j(A_n))(\exp(d_j(\{e\})) - 1)$$

$$d_j(B_n) \geq d_j(A_n)$$

which is true for all $A_n \subseteq B_n$ and elements $e \in [K_+] \setminus B_n$.

For $r_{nj} = 1$ we have the following similar inequality:

$$\frac{1 + \exp\left(-d_j(A_n)\right)}{1 + \exp\left(-d_j(A_n \cup \{e\})\right)} \geq \frac{1 + \exp\left(-d_j(B_n)\right)}{1 + \exp\left(-d_j(B_n \cup \{e\})\right)}$$

and after similar algebra as the $r_{nj} = 0$ case, we obtain the same final inequality

$$d_j(B_n) \geq d_j(A_n) \tag{5.16}$$

which is again true for all $A_n \subseteq B_n$ and elements $e \in [K_+] \setminus B_n$. Therefore the pILA likelihood, and in turn, the pILA joint probability is submodular for each observation's feature assignments. □

5.3 Leaky, Noisy-Or Binary Data Model

In this section, we consider the following model for $N \times D$ binary data $\boldsymbol{X}$:

$$\boldsymbol{Z}|\alpha, \beta \sim \text{IBP}(\alpha, \beta) \tag{5.17}$$

$$\boldsymbol{A}|\theta_1 \sim \text{H}_1(\theta_1) \tag{5.18}$$

$$\lambda_d|\theta_2 \sim \text{H}_2(\theta_2) \tag{5.19}$$

$$x_{nd}|\epsilon, \boldsymbol{Z}_{n\cdot}, \boldsymbol{A}, \lambda_d \sim \text{Bernoulli}(1 - \epsilon\lambda_d^{\boldsymbol{Z}_{n\cdot}\boldsymbol{A}_{\cdot d}}) \tag{5.20}$$

where $\text{H}_i(\theta_i), i \in \{1, 2\}$ are nonnegative distributions where the support of H_2 must complement the parameter ϵ so that the product $\epsilon\lambda_d^{\boldsymbol{Z}_{n\cdot}\boldsymbol{A}_{\cdot d}}$ takes values in a subset of $(0, 1)$, and $\boldsymbol{A} \in \mathbb{R}_+^{K_+ \times D}$ is a $K_+ \times D$ nonnegative matrix.

Again, to show the log-submodularity of this model, we only need to show that the likelihood is a log-submodular function of each observation's feature assignments because the IBP prior is log submodular and the other probabilistic components only depend on $\boldsymbol{Z}$ through the K_+ selection. The log-likelihood is

$$\sum_{n=1}^{N}\sum_{d=1}^{D}\left[x_{nd}\log\left(1 - \epsilon\lambda_d^{\boldsymbol{Z}_{n\cdot}\boldsymbol{A}_{\cdot d}}\right) + (1 - x_{nd})\log\left(\epsilon\lambda_d^{\boldsymbol{Z}_{n\cdot}\boldsymbol{A}_{\cdot d}}\right)\right]. \tag{5.21}$$

As a function of each observation's feature assignment, the log-likelihood is

$$\mathcal{F}(\boldsymbol{Z}_{n\cdot}) = \sum_{d=1}^{D} \left[x_{nd} \log \left(1 - \epsilon \lambda_d^{\boldsymbol{Z}_{n\cdot} \boldsymbol{A}_{\cdot d}} \right) + (1 - x_{nd}) \log \left(\epsilon \lambda_d^{\boldsymbol{Z}_{n\cdot} \boldsymbol{A}_{\cdot d}} \right) \right]. \tag{5.22}$$

Since nonnegative linear combinations of submodular functions are submodular, we can prove the submodularity of $\mathcal{F}(\boldsymbol{Z}_{n\cdot})$ by proving the submodularity of $\mathcal{H}_d(\boldsymbol{Z}_{n\cdot})$ for an arbitrary but specific d where

$$\mathcal{F}(\boldsymbol{Z}_{n\cdot}) = \sum_{d=1}^{D} \mathcal{H}_d(\boldsymbol{Z}_{n\cdot}). \tag{5.23}$$

Similar to the previous sections, we use the definition of submodularity and the equivalent set function

$$\mathcal{H}_d(\boldsymbol{Z}_{n\cdot}) = \mathcal{G}_d(A_n) = x_{nd} \log \left(1 - \epsilon \lambda_d^{\sum_{k \in A_n} a_{kd}} \right) + (1 - x_{nd}) \log \left(\epsilon \lambda_d^{\sum_{k \in A_n} a_{kd}} \right) \tag{5.24}$$

Again, we handle $x_{nd} = 0$ and $x_{nd} = 1$ case separately. For $x_{nd} = 0$, the definition of submodularity yields the following inequality for $A_n \subseteq B_n \subseteq [K_+]$ and element $e \in [K_+] \setminus B_n$:

$$\log \left(\frac{\epsilon \lambda_d^{\sum_{k \in A_n \cup \{e\}} a_{kd}}}{\epsilon \lambda_d^{\sum_{k \in A_n} a_{kd}}} \right) \geq \log \left(\frac{\epsilon \lambda^{\sum_{k \in B_n \cup \{e\}} a_{kd}}}{\epsilon \lambda_d^{\sum_{k \in B_n} a_{kd}}} \right) \tag{5.25}$$

$$\log \left(\frac{\epsilon \lambda_d^{a_{ed}}}{\epsilon \lambda_d^{a_{ed}}} \right) \geq \log \left(\frac{\epsilon \lambda_d^{a_{ed}}}{\epsilon \lambda_d^{a_{ed}}} \right) \tag{5.26}$$

$$0 \geq 0 \tag{5.27}$$

which is true for all $A_n \subseteq B_n$ and elements $e \in [K_+] \setminus B_n$. For $x_{nd} = 1$, we have

the following inequality:

$$\log\left(\frac{1-\epsilon\lambda_d^{\sum_{k\in A_n\cup\{e\}}a_{kd}}}{1-\epsilon\lambda_d^{\sum_{k\in A_n}a_{kd}}}\right) \geq \log\left(\frac{1-\epsilon\lambda_d^{\sum_{k\in B_n\cup\{e\}}a_{kd}}}{1-\epsilon\lambda_d^{\sum_{k\in B_n}a_{kd}}}\right) \tag{5.28}$$

$$\frac{1-\epsilon\lambda_d^{\sum_{k\in A_n\cup\{e\}}a_{kd}}}{1-\epsilon\lambda_d^{\sum_{k\in A_n}a_{kd}}} \geq \frac{1-\epsilon\lambda_d^{\sum_{k\in B_n\cup\{e\}}a_{kd}}}{1-\epsilon\lambda_d^{\sum_{k\in B_n}a_{kd}}} \tag{5.29}$$

where cross-multiplying and cancelling the common terms yields

$$\lambda_d^{\sum_{k\in B_n}a_{kd}+a_{ed}} + \lambda_d^{\sum_{k\in A_n}a_{kd}} \geq \lambda_d^{\sum_{k\in A_n}a_{kd}+a_{ed}} + \lambda_d^{\sum_{k\in B_n}a_{kd}} \tag{5.30}$$

$$\lambda_d^{\sum_{k\in B_n}a_{kd}+a_{ed}} - \lambda_d^{\sum_{k\in B_n}a_{kd}} \geq \lambda_d^{\sum_{k\in A_n}a_{kd}+a_{ed}} - \lambda_d^{\sum_{k\in A_n}a_{kd}} \tag{5.31}$$

$$\lambda_d^{\sum_{k\in B_n}a_{kd}}(\lambda_d^{a_{ed}}-1) \geq \lambda_d^{\sum_{k\in A_n}a_{kd}}(\lambda_d^{a_{ed}}-1) \tag{5.32}$$

$$\lambda_d^{\sum_{k\in B_n}a_{kd}} \geq \lambda_d^{\sum_{k\in A_n}a_{kd}} \tag{5.33}$$

$$\sum_{k\in B_n} a_{kd} \geq \sum_{k\in A_n} a_{kd} \tag{5.34}$$

which is true for all $A_n \subseteq B_n$ and elements $e \in [K_+] \setminus B_n$. Therefore the leaky, noisy-or model is submodular for each observation's feature assignment. □

5.4 Inference Techniques

In Chapter 4, we performed inference on a linear-Gaussian IBP model by iteratively using submodular maximization to obtain the latent feature assignments and variational updates to maintain the model parameters via the maximization-expectation (ME) framework. The ME framework can be applied to more complicated (non-conjugate) feature models as well, e.g. pILA, but the variational updates for the model parameters will not have a closed-form solution.

Non-conjugate variational inference is a well-studied topic and several methods could be applied to a given model. For instance, Wang and Blei [2012b] proposed "Laplace" and "delta-method" variational inference algorithms that could be applicable to models with any exponential-family likelihood: the Laplace method effectively places a variational Gaussian distribution at the MAP estimate of the model parameters while the delta-method optimizes a Taylor expansion of

the variational lower bound. Other nonconjugate variational methods that could be used in an ME setting include those proposed in: Knowles and Minka [2011], Gershman *et al.* [2012], Honkela and Valpola [2004], and Ahmed and Xing [2007].

A simpler approach as ME inference, is to compute MAP estimates of the model parameters as well—the so-called "maximization-maximization" or "iterated-conditional-modes" technique. This method can lead to simple optimization routines for the model parameters in addition to the submodular optimization for the feature assignments. For instance, with the pILA model, if we choose a log-normal prior on the affinity weights then updating each weight can be phrased as a convex minimization.[1]

Another simplifying possibility, would be to form objective functions by taking 0-variance asymptotic limits of the probabilistic model. Broderick *et al.* [2013a] took the 0-variance asymptotic limit of the linear-Gaussian IBP model discussed in Chapter 4. From this limit, the authors derived a simple optimization scheme where each individual feature z_{nk} was optimized independently and the model parameters were set to their expected value conditioned on the feature assignments, see §4.8. By using a nonnegative prior on the model parameters, this same 0-variance asymptotic limit could be used to obtain an objective function that is submodular for each observation's feature assignments. We conjecture that we could obtain submodular 0-variance asymptotic objective functions for a number of latent feature models but leave this investigation for future research.

[1] We state this result without proof as we are currently exploring this model and will formalize these results in a future publication.

Chapter 6

Conclusions and Future Work

This thesis established a novel connection between the field of Bayesian nonparametrics and submodular optimization. We showed that the combinatorial structures that arise from nonparametric feature allocation models often demonstrate submodularity. This observation allowed us to draw upon the theoretical benefits and wealth of methodology developed for submodular optimization in order to perform MAP inference with nonparametric feature allocation models. For the linear-Gaussian feature allocation model, we showed that submodular maximization can lead to fast MAP estimates of the feature allocation that capture the structure of held-out data as well as the best sampling/variational inference techniques. Our methodology (the MEIBP algorithm) and experimental results demonstrated that it is possible to perform inference with large data and highly combinatorial Bayesian nonparametric models, e.g. we performed inference for a latent feature allocation model using a dataset with 10^8 total observations in circa ten hours.

While this thesis explicitly demonstrated the utility of submodular optimization in Bayesian nonparametric models, it has also implicitly provided a new application domain and motivation for researchers studying submodular optimization. Indeed, the specific flavor of submodular optimization studied in this thesis—unconstrained submodular maximization—is currently a very active area of research in submodular optimization. Buchbinder *et al.* [2012] recently showed that a linear stochastic algorithm can obtain an expected (1/2)-approximation for unconstrained submodular maximization problems, and Iyer *et al.* [2013] showed a

general unifying framework for submodular optimization problems that subsumes the work of Buchbinder *et al.* [2012]. As demonstrated in this thesis, advances in the field of submodular optimization will benefit the Bayesian nonparametrics community. For instance, whether a deterministic algorithm can obtain a (1/2)-approximation for unconstrained submodular problems is an open question, and a positive result would provide a possibly improved MEIBP algorithm.

There are several directions for future research based upon the contributions from this thesis:

- *design and implement submodular-based MAP inference for additional feature allocation models*: Certain feature allocation models, such as the Infinite Latent Attribute Model (see §5.2), exhibit submodularity and can not scale to sizeable datasets via sampling-based inference, e.g. datasets with more than one thousand observations. One key direction for future research is to design and implement submodular-maximization-based inference algorithms so that these models can be used on large datasets.

- *generalization of submodularity results*: The distribution specified by the IBP can be obtained by integrating a beta process random measure with respect to a Bernoulli process de Finetti mixing measure [Thibaux and Jordan, 2007]. An interesting question is whether the submodularity results from this thesis can be stated in terms of the underlying random measures, e.g. whether the submodularity result can take a more general form based on properties of the beta-Bernoulli process.

- *find submodularity in other nonparametric or deep models*: There are many discrete Bayesian nonparametric processes, such as the Dirichlet process and beta negative binomial process, and an interesting area for future research will be to generalize our results in order to phrase inference with models that embed these processes as submodular optimization problems.

- *characterize where unbounded nonparametric models are useful*: As we saw in §4.9, the the bounded MEIBP and variational approaches, truncated samplers, and parametric BNMF model performed as-well or better than the unbounded samplers in several experiments. Formally exploring and

characterizing where parametric, bounded, and unbounded models succeed is an interesting, and understudied, question in Bayesian nonparametrics.

Appendix A

A.1 Truncated Gaussian Properties

In the §4.1 we examined a truncated Gaussian of the form:

$$\mathcal{TN}(\tilde{\mu}_{kd}, \tilde{\sigma}^2_{kd}) = \frac{2}{\operatorname{erfc}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right)} \mathcal{N}(\tilde{\mu}_{kd}, \tilde{\sigma}^2_{kd}) \tag{A.1}$$

with $\mathcal{N}$ representing a Gaussian distribution. The first two moments of $\mathcal{TN}(\tilde{\mu}_{kd}, \tilde{\sigma}^2_{kd})$ are:

$$\mathbb{E}\left[a_{kd}\right] = \tilde{\mu}_{kd} + \tilde{\sigma}_{kd} \frac{\sqrt{2/\pi}}{\operatorname{erfcx}\left(\wp_{kd}\right)} \tag{A.2}$$

$$\mathbb{E}\left[a^2_{kd}\right] = \tilde{\mu}^2_{kd} + \tilde{\sigma}^2_{kd} + \tilde{\sigma}_{kd}\tilde{\mu}_{kd} \frac{\sqrt{2/\pi}}{\operatorname{erfcx}\left(\wp_{kd}\right)} \tag{A.3}$$

with $\wp_{kd} = -\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}$ and $\operatorname{erfcx}(y) = e^{y^2}(1 - \operatorname{erf}(y))$ representing the scaled complementary error function. The entropy is

$$\begin{aligned} H(q(a_{kd})) =& \frac{1}{2} \ln \frac{\pi e \tilde{\sigma}^2_{kd}}{2} + \ln \operatorname{erfc}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right) \\ &+ \frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}} \sqrt{\frac{1}{2\pi}} \left(\operatorname{erfcx}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right)\right)^{-1}. \end{aligned} \tag{A.4}$$

A.2 Submodular Joint log-IBP Distribution Counter Example.

In this section we show that the log of the joint IBP distribution is not jointly submodular. Intuitively, this is because the IBP distribution favors assigning either a large or small number of observations to each feature, which does not coincide with the "diminishing returns" characteristic of submodular functions. Here's a concrete counter example using the single parameter IBP with shifted equivalence classes:

$$\log\left(P([\boldsymbol{Z}]_{\text{shift}})\right) = \log\left(\frac{\alpha^{K_+}}{K_+!}\mathrm{e}^{-\alpha H_N}\right) + \sum_{k=1}^{K^+}\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right),$$

where $\alpha > 0$ is a parameters, K_+ is the number of active latent features, H_N is the N^{th} harmonic number, N is the number of observations, and m_k is the number of observations that are assigned the k^{th} feature.

Note: defining $\log\left(P([\boldsymbol{Z}]_{\text{shift}})\right)$ as a set function is a bit more tortuous than the conditional case since we are working with a binary matrix instead of a binary vector, so our set function is defined over a product space $N \times \mathbb{N}$, where $\mathbb{N}$ are the natural numbers. Fortuneately, we can work directly with the binary matrix $\boldsymbol{Z}$, while keeping in mind that we are actually working with an equivalent set defined over a product space. For instance, if a binary matrix has the third and fifth feature active for the sixth observarion, the equivalent set defined over the product space would be: $\{(6,3),(6,5)\}$. The subsets of this matrix are: $\{(6,3),(6,5)\}$ or $\{(6,3)\}$ or $\{(6,5)\}$ or $\{\}$.

For this counter example let $K_+ = 1$ and examine binary matrix $\boldsymbol{Z}_A$ where $\text{set}(\boldsymbol{Z}_A) = A = \{(1,1)\}$ which is a subset of $\boldsymbol{Z}_B$ where

$$\text{set}(\boldsymbol{Z}_B) = B = \{(1,1),(1,2),\ldots,(1,N-1)\}.$$

If $\log\left(P([\boldsymbol{Z}]_{\text{shift}})\right)$ is submodular then

$$\log\left(P([\boldsymbol{Z}_{A\cup(1,N)}])\right) - \log\left(P([\boldsymbol{Z}_A])\right) \geq \log\left(P([\boldsymbol{Z}_{B\cup(1,N)}])\right) - \log\left(P([\boldsymbol{Z}_B])\right)$$

where $m_1 = 2$ for $A\cup(1,N)$, $m_1 = 1$ for A, $m_1 = N$ for $B\cup(1,N)$, and $m_1 = N-1$ for B. Since K_+ does not change, all terms in the IBP definition cancel except the $\log\left((N-m_k)!(m_k-1)!\right)$ terms, yielding the inequality:

$$\begin{aligned}&\log\left((N-2)!(2-1)!\right) - \log\left((N-1)!(1-1)!\right)\\ &\quad\geq \log\left((N-N)!(N-1)!\right) - \log\left((N-(N-1))!((N-1)-1)!\right)\end{aligned}$$

which simplifies to:

$$-\log\left(N-1\right) \geq \log\left(N-1\right)$$

which is false for all $N > 2$ and $\log\left(P([\boldsymbol{Z}])\right)$. As a result, the joint IBP distribution is not submodular. □

A.3 Nonnegative Linear-Gaussian Derivations

A.3.1 Evidence Lower Bound

Given the nonnegative linear Gaussian model described in §4.1, the evidence lower bound is

$$\mathcal{L} = \mathbb{E}_q[\log\, p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\theta})] + H[q] \tag{A.5}$$

$$= \mathbb{E}_q[\log\, p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{A}, \sigma_{\boldsymbol{X}}^2)] + \mathbb{E}_q[\log\, p(\boldsymbol{A}|\sigma_{\boldsymbol{A}}^2)] + \mathbb{E}_q[\log\, p([\boldsymbol{Z}]|\alpha)] + H[q(\boldsymbol{A})] + H[q(\boldsymbol{Z})]. \tag{A.6}$$

Since we use $q(\boldsymbol{Z}) = \delta(\boldsymbol{Z} - \boldsymbol{Z}^*)$, all $\mathbb{E}_q[\cdot] \equiv \mathbb{E}_{q(\boldsymbol{A})q(\boldsymbol{Z})} \equiv \mathbb{E}_{q(\boldsymbol{A})}$ and all $\boldsymbol{Z}$ occurrences go to $\boldsymbol{Z}^*$. Therefore we use the shorthand $\mathbb{E}[\cdot]$ to indicate $\mathbb{E}_{q(\boldsymbol{A})}$.

As mentioned in §4.1, we overload our notation and let $\boldsymbol{Z} \equiv \boldsymbol{Z}^*$ for the variational lower bound because throughout this work we examine the variational lower bound as the objective function to find $\boldsymbol{Z}^*$. We only use the actual value of $\boldsymbol{Z}^*$ when evaluating the variational lower bound, i.e. to test the inference implementation.

For the Gaussian likelihood component, we have

$$\begin{aligned}
&\mathbb{E}[\log p(\boldsymbol{X}_{n\cdot}|\boldsymbol{Z}, \boldsymbol{A}, \sigma_{\boldsymbol{X}}^2)] \\
&\quad = \mathbb{E}[-\frac{D}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) - \frac{1}{2\sigma_{\boldsymbol{X}}^2}(\boldsymbol{X}_{n\cdot} - \boldsymbol{Z}_{n\cdot}\boldsymbol{A})(\boldsymbol{X}_{n\cdot} - \boldsymbol{Z}_{n\cdot}\boldsymbol{A})^T] \\
&\quad = -\frac{D}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) - \frac{1}{2\sigma_{\boldsymbol{X}}^2}\left(\boldsymbol{X}_{n\cdot}\boldsymbol{X}_{n\cdot}^T - 2\boldsymbol{Z}_{n\cdot}\mathbb{E}[\boldsymbol{A}]\boldsymbol{X}_{n\cdot}^T + \mathbb{E}[\boldsymbol{Z}_{n\cdot}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{Z}_{n\cdot}]\right)
\end{aligned}$$

where

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{Z}_{n\cdot}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{Z}_{n\cdot}^T] = \\
&\quad \mathbb{E}\Big[\sum_{d=1}^{D}\sum_{k=1}^{K} z_{nk}^2 a_{kd}^2 + \sum_{k':k'\neq k} z_{nk}z_{nk'}a_{kd}a_{k'd}\Big] \\
&\quad = \sum_{d=1}^{D}\sum_{k=1}^{K} z_{nk}\mathbb{E}[a_{kd}^2] + \sum_{k':k'\neq k} z_{nk}z_{nk'}\mathbb{E}[a_{kd}]\mathbb{E}[a_{k'd}] \\
&\quad = \sum_{k=1}^{K} z_{nk}\sum_{d=1}^{D}\mathbb{E}[a_{kd}^2] + \sum_{k=1}^{K}\sum_{k':k'\neq k} z_{nk}z_{nk'}\sum_{d=1}^{D}\mathbb{E}[a_{kd}]\mathbb{E}[a_{k'd}] \\
&\quad = \sum_{k=1}^{K} z_{nk}\sum_{d=1}^{D}\mathbb{E}[a_{kd}^2] + \boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T - \sum_{k=1}^{K} z_{nk}\sum_{d=1}^{D}\mathbb{E}[a_{kd}]^2 \\
&\quad = \boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \sum_{k=1}^{K} z_{nk}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}^2] - \mathbb{E}[a_{kd}]^2\right]
\end{aligned}$$

where $\boldsymbol{\Phi}$ is a matrix with entries $\phi_{kd} = \mathbb{E}\left[a_{kd}\right]$. Combining this result with all

likelihood terms yields

$$
\begin{aligned}
&\mathbb{E}[\log p(\boldsymbol{X}_{n\cdot}|\boldsymbol{Z},\boldsymbol{A},\sigma_{\boldsymbol{X}}^2)] \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\left(\boldsymbol{X}_{n\cdot}\boldsymbol{X}_{n\cdot}^T - 2\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{X}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \sum_{k=1}^{K} z_{nk}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}^2] - \mathbb{E}[a_{kd}]^2\right]\right) \\
&\quad - \frac{D}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) \\
&= \frac{1}{\sigma_{\boldsymbol{X}}^2}\left(-\frac{1}{2}\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot}\boldsymbol{\xi}_{n\cdot}^T - \frac{1}{2}\boldsymbol{X}_{n\cdot}\boldsymbol{X}_{n\cdot}^T\right) - \frac{D}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right)
\end{aligned}
$$

with

$$
\xi_{nk} = \boldsymbol{\Phi}_{k\cdot}\boldsymbol{X}_{n\cdot}^T + \frac{1}{2}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}]^2 - \mathbb{E}[a_{kd}^2]\right]. \tag{A.7}
$$

The variational lower bound component for the zero-mean, truncated Gaussian latent factors is:

$$
\begin{aligned}
\mathbb{E}_q[\log p(a_{kd}|\alpha)] &= \mathbb{E}\left[\log(2) - \frac{1}{2}\log\left(2\pi\sigma_{\boldsymbol{A}}^2\right) - \frac{1}{2\sigma_{\boldsymbol{A}}^2}a_{kd}^2\right] \\
&= \log(2) - \frac{1}{2}\log\left(2\pi\sigma_{\boldsymbol{A}}^2\right) - \frac{1}{2\sigma_{\boldsymbol{A}}^2}\mathbb{E}\left[a_{kd}^2\right]
\end{aligned}
$$

The variational lower bound component for the IBP prior is simply the log of the prior:

$$
\mathbb{E}_q[\log p(\boldsymbol{Z}|\alpha)] = K_+\log(\alpha) - \log(K_+!) - \alpha H_N + \sum_{k=1}^{K^+}\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right)
$$

The entropy for latent factor variational distribution is the entropy for a trun-

cated Gaussian:

$$H(q(a_{kd})) = \frac{1}{2}\log\left(\frac{\pi e\tilde{\sigma}_{kd}^2}{2}\right) + \log\left(\text{erfc}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right)\right) + \frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}}\sqrt{\frac{1}{2\pi}}\left(\text{erfcx}\left(-\frac{\tilde{\mu}_{kd}}{\tilde{\sigma}_{kd}\sqrt{2}}\right)\right)^{-1}.$$

The entropy for the latent feature assignment, $H(q(Z)) = H(\delta(Z - Z^*))$ is zero since it is a point assignment for a discrete distribution.

Combining the above terms results in the following variational lower bound:

$$\begin{aligned}\mathcal{L} = &\sum_{n=1}^{N}\Big[\frac{1}{\sigma_{\boldsymbol{X}}^2}\left(-\frac{1}{2}\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot}\boldsymbol{\xi}_{n\cdot}^T - \frac{1}{2}\boldsymbol{X}_{n\cdot}\boldsymbol{X}_{n\cdot}^T\right) - \frac{D}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right)\Big] \\ &+ \sum_{d=1}^{D}\sum_{k=1}^{K}\Big[\log(2) - \frac{1}{2}\log\left(2\pi\sigma_{\boldsymbol{A}}^2\right) - \frac{1}{2\sigma_{\boldsymbol{A}}^2}\mathbb{E}\left[a_{kd}^2\right]\Big] \\ &+ K_+\log(\alpha) - \log(K_+!) - \alpha H_N + \sum_{k=1}^{K^+}\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right) \\ &+ \sum_{d=1}^{D}\sum_{k=1}^{K}H[q(a_{kd})]\end{aligned}$$

Some algebraic rearrangements yield

$$\begin{aligned}\mathcal{L} = &\frac{1}{\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[-\frac{1}{2}\boldsymbol{Z}_{n\cdot}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot}\boldsymbol{\xi}_{n\cdot}^T\right] - \log(K_+!) \\ &+ \sum_{k=1}^{K^+}\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right) + \sum_{k=1}^{K}\eta_k \\ &- \frac{1}{2\sigma_{\boldsymbol{X}}}trace\left(\boldsymbol{X}\boldsymbol{X}^T\right) - \frac{ND}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) - \alpha H_N\end{aligned}$$

with

$$\eta_k = \frac{1}{2}\sum_{d=1}^{D}\Big[-\log\left(\frac{\pi\sigma_{\boldsymbol{A}}^2}{2\alpha^{2/D}}\right) - \frac{\mathbb{E}[a_{kd}^2]}{\sigma_{\boldsymbol{A}}^2} + 2H(q(a_{kd}))\Big],$$

which is equivalent to the evidence lower bound presented in §4.1, where

$$-\frac{1}{2\sigma_{\boldsymbol{X}}}trace\left(\boldsymbol{X}\boldsymbol{X}^T\right) - \frac{ND}{2}\log\left(2\pi\sigma_{\boldsymbol{X}}^2\right) - \alpha H_N$$

is replaced with "constant."[1]

A.3.2 Hyperparameter Inference

In the inference procedure discussed in Chapter 4, text we assumed the hyperparameters $\boldsymbol{\theta} = \{\sigma_{\boldsymbol{X}}, \sigma_{\boldsymbol{A}}, \alpha\}$ were known (i.e. estimated from the data). Placing conjugate gamma hyperpriors on these parameters allows for a straightforward extension in which we infer their values. Formally, let

$$p(\tau_X) = \text{Gamma}(\tau_X; a_X, b_X) \tag{A.8}$$
$$p(\tau_A) = \text{Gamma}(\tau_A; a_A, b_A) \tag{A.9}$$
$$p(\alpha) = \text{Gamma}(\alpha; a_\alpha, b_\alpha) \tag{A.10}$$

where τ represents the precision, equivalent to the inverse variance $\frac{1}{\sigma^2}$, for the variance parameter indicated in the subscript. Update equations for the variational distributions follow from standard update equations for variational inference in

[1] As discussed in Appendix A.3.2, performing variational inference on the hyperparameters would reinsert these three terms into the variational lower bound, i.e. they would not be constant. However, these terms would still not be included in the $\boldsymbol{Z}$ optimization.

exponential families, cf. Attias [2000], and yield:

$$q(\tau_X) = \text{Gamma}(\widetilde{a}_X, \widetilde{b}_X) \tag{A.11}$$
$$q(\tau_A) = \text{Gamma}(\widetilde{a}_A, \widetilde{b}_A) \tag{A.12}$$
$$q(\alpha) = \text{Gamma}(\widetilde{a}_\alpha, \widetilde{b}_\alpha) \tag{A.13}$$

with variance updates

$$\widetilde{a}_A = a_A + \frac{KD}{2} \tag{A.14}$$
$$\widetilde{b}_A = b_A + \frac{1}{2}\sum_{k=1}^{K_+}\sum_{d=1}^{D}\mathbb{E}\left[a_{kd}^2\right] \tag{A.15}$$

and

$$\widetilde{a}_X = a_X + \frac{ND}{2} \tag{A.16}$$
$$\widetilde{b}_X = b_X + \frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D}\Big[x_{nd}^2 + \sum_{k=1}^{K_+}\Big[\mathbb{E}\left[a_{kd}^2\right] z_{nk} \tag{A.17}$$
$$- 2\mathbb{E}[a_{kd}]z_{nk}x_{nd} + 2\sum_{k'=k+1}^{K_+} z_{nk}z_{nk'}a_{kd}a_{k'd}\Big]\Big] \tag{A.18}$$

and $q(\alpha)$ updates

$$\widetilde{a}_\alpha = a_\alpha + K_+ \tag{A.19}$$
$$\widetilde{b}_\alpha = b_\alpha + H_N. \tag{A.20}$$

MEIBP inference is carried out exactly as discussed in Chapter 4 except all instances of $\sigma_{\boldsymbol{X}}, \sigma_{\boldsymbol{A}}$, and α are replaced with the expectations from their respective variational distribution. Note that some of the "const" terms would no longer be constant, but these terms would not affect the $\boldsymbol{Z}$ optimization results. Furthermore the variational lower bound also has three additional entropy terms for

gamma distributions, one for each hyperparameter.

A.3.3 Variational Updates for $q(\boldsymbol{A})$

Following standard mean field variational Bayes updates [Attias, 2000; Ghahramani and Beal, 2001], we can compute the variational updates as follows:

$$
\begin{aligned}
\log\left(q(a_{kd})\right) &= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{A}_{-kd}}[\log\left(p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{A}|\Theta)\right)] + const \\
&= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{A}_{-kd}}[\log\left(p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{A},\Theta)\right)] + \mathbb{E}_{\boldsymbol{Z},\boldsymbol{A}_{-kd}}[\log\left(p(\boldsymbol{A}|\Theta)\right)] + const \\
&= \mathbb{E}_{\boldsymbol{A}_{-kd}}[\log\left(p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{A},\Theta)\right)] + \mathbb{E}_{\boldsymbol{A}_{-kd}}[\log\left(p(\boldsymbol{A}|\Theta)\right)] + const \\
&= \mathbb{E}_{\boldsymbol{A}_{-kd}}\left[\sum_{n=1}^{N} -\frac{1}{2\sigma_{\boldsymbol{X}}^2}(\boldsymbol{X}_{n\cdot} - \boldsymbol{Z}_{n\cdot}\boldsymbol{A})(\boldsymbol{X}_{n\cdot} - \boldsymbol{Z}_{n\cdot}\boldsymbol{A})^T\right] - \frac{a_{kd}^2}{2\sigma_{\boldsymbol{A}}^2} + const \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{A}_{-kd}}\left[\boldsymbol{Z}_{n\cdot}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{Z}_{n\cdot} - 2\boldsymbol{Z}_{n\cdot}\boldsymbol{A}\boldsymbol{X}_{n\cdot}^T\right] - \frac{a_{kd}^2}{2\sigma_{\boldsymbol{A}}^2} + const \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[\mathbb{E}_{\boldsymbol{A}_{-kd}}\left[\boldsymbol{Z}_{n\cdot}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{Z}_{n\cdot}\right] - 2a_{kd}z_{nk}x_{nd}\right] - \frac{a_{kd}^2}{2\sigma_{\boldsymbol{A}}^2} + const \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[\mathbb{E}_{\boldsymbol{A}_{-kd}}\left[a_{kd}^2 z_{nk} + a_{kd}z_{nk}2\sum_{k'\neq k} z_{nk'}a_{k'd}\right] - 2a_{kd}z_{nk}x_{nd}\right] - \frac{a_{kd}^2}{2\sigma_{\boldsymbol{A}}^2} + const \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[a_{kd}^2 z_{nk} + a_{kd}z_{nk}2\sum_{k'\neq k} z_{nk'}\mathbb{E}\left[a_{k'd}\right] - 2a_{kd}z_{nk}x_{nd}\right] - \frac{a_{kd}^2}{2\sigma_{\boldsymbol{A}}^2} + const \\
&= -\frac{1}{2\sigma_{\boldsymbol{X}}^2}\left[a_{kd}^2(m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}) + 2a_{kd}\sum_{n=1}^{N}\left(z_{nk}\sum_{k'\neq k} z_{nk'}\mathbb{E}\left[a_{k'd}\right] - z_{nk}x_{nd}\right)\right] + const \\
&= -\frac{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[a_{kd}^2 + \frac{2a_{kd}}{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}}\sum_{n=1}^{N} z_{nk}\left(\sum_{k'\neq k} z_{nk'}\mathbb{E}\left[a_{k'd}\right] - x_{nd}\right)\right] + const \\
&= -\frac{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}}{2\sigma_{\boldsymbol{X}}^2}\sum_{n=1}^{N}\left[a_{kd}^2 - 2\frac{a_{kd}}{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}}\sum_{n=1}^{N} z_{nk}\left(x_{nd} - \sum_{k'\neq k} z_{nk'}\mathbb{E}\left[a_{k'd}\right]\right)\right] + const
\end{aligned}
$$

which shows that $q(a_{kd})$ is a truncated Gaussian distribution with

$$\tilde{\mu}_{kd} = \frac{1}{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}} \sum_{n=1}^{N} z_{nk} \left(x_{nd} - \sum_{k' \neq k} z_{nk'} \mathbb{E}\left[a_{k'd}\right] \right) \tag{A.21}$$

$$\tilde{\sigma}_{kd}^2 = \frac{\sigma_X^2}{m_k + \frac{\sigma_{\boldsymbol{X}}^2}{\sigma_{\boldsymbol{A}}^2}} \tag{A.22}$$

A.3.4 Evidence as a function of $\mathbf{Z}_{n\cdot}$

As shown in Chapter 4, we obtain a submodular objective function for each $\boldsymbol{Z}_{n\cdot}$, $n \in \{1, \ldots, N\}$ by examining the evidence as a function of $\boldsymbol{Z}_{n\cdot}$ while holding constant all $n' \in \{1, \ldots, N\} \setminus n$. The evidence is

$$\begin{aligned} &\frac{1}{\sigma_{\boldsymbol{X}}^2} \sum_{n=1}^{N} \left[-\frac{1}{2} \boldsymbol{Z}_{n\cdot} \boldsymbol{\Phi} \boldsymbol{\Phi}^T \boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot} \boldsymbol{\xi}_{n\cdot}^T \right] - \log\left(K_+!\right) \\ &\quad + \sum_{k=1}^{K^+} \left[\log\left(\frac{(N - m_k)!(m_k - 1)!}{N!} \right) + \eta_k \right] + \text{const} \end{aligned} \tag{A.23}$$

$$\xi_{nk} = \boldsymbol{\Phi}_{k\cdot} \boldsymbol{X}_{n\cdot}^T + \frac{1}{2} \sum_{d=1}^{D} \left[\mathbb{E}[a_{kd}]^2 - \mathbb{E}[a_{kd}^2] \right] \tag{A.24}$$

$$\eta_k = \sum_{d=1}^{D} \left[-\frac{\log\left(\frac{\pi \sigma_{\boldsymbol{A}}^2}{2\alpha^{2/D}} \right)}{2} - \frac{\mathbb{E}[a_{kd}^2]}{2\sigma_{\boldsymbol{A}}^2} + H(q(a_{kd})) \right], \tag{A.25}$$

which nearly factorizes over the $\boldsymbol{Z}_{n\cdot}$ because the likelihood component and parts of the prior components naturally fit into a quadratic function of $\boldsymbol{Z}_{n\cdot}$. The $\log K_+!$ and η_k only couple the rows of $\boldsymbol{Z}$ when K_+ changes, while the log-factorial term couples the rows of $\boldsymbol{Z}$ through the sums of the columns. Both of these terms only depend on statistics of $\boldsymbol{Z}$ (the m_k values and K_+), not the $\boldsymbol{Z}$ matrix itself, e.g. permuting the rows of $\boldsymbol{Z}$ would not affect these terms. Furthermore, $\log\left(K_+\right)$ and η_k have no N dependence and become insignificant as N increases. These observations, in conjunction with the MEIBP performance in the experimental section of Chapter 4, indicate that sequentially optimizing Eq. A.23 for $\boldsymbol{Z}_{n\cdot}$ is a

reasonable surrogate for optimizing $\boldsymbol{Z}$.

Here we explicitly decompose Eq. A.23 to show its $\boldsymbol{Z}_{n\cdot}$ dependency. Decomposing $\log\left(\frac{(N-m_k)!(m_k-1)!}{N!}\right)$ is straightforward if we first define the function:

$$\nu(z_{nk}) = \begin{cases} \log\left((N - m_{k\backslash n} - z_{nk})!(m_{k\backslash n} + z_{nk} - 1)!/N!\right) \\ 0, \quad \text{if } m_{k\backslash n} = 0 \text{ and } z_{nk} = 0. \end{cases} \tag{A.26}$$

where the "$\backslash n$" subscript indicates the variable with the n^{th} row removed from $\boldsymbol{Z}$. For a given n we have:

$$\begin{aligned} \sum_{k=1}^{K_+} \nu(z_{nk}) &= \sum_{k=1}^{K_+} \log\left((N - m_k)!(m_k - 1)!/N!\right) \\ &= \sum_{k=1}^{K_+} z_{nk}\left(\nu(z_{nk} = 1) - \nu(z_{nk} = 0)\right) \\ &\quad + \nu(z_{nk} = 0), \end{aligned} \tag{A.27}$$

which makes the $\boldsymbol{Z}_{n\cdot}$ dependency explicit and lets us add $\nu(z_{nk} = 1) - \nu(z_{nk} = 0)$ into the inner-product term, $\boldsymbol{\xi}_{n\cdot}$, and place $\nu(z_{nk} = 0)$ into a constant term. We can incorporate η_k into the inner-product term in a similar manner for a given $n \in \{1, \ldots, N\}$:

$$\sum_{k=1}^{K_+} \eta_k = \sum_{k:m_{k\backslash n}>0} \eta_k + \sum_{k=1}^{K_+} \mathbf{1}_{\{m_{k\backslash n}=0\}} z_{nk}\eta_k, \tag{A.28}$$

where the first term does not depend on $\boldsymbol{Z}_{n\cdot}$ and is added to the constant term, while the second term is added to the inner-product term. Finally, for a given $n \in \{1, \ldots, N\}$ the $\log(K!)$ term becomes

$$\log(K_+!) = \log\left(\left(K_{+\backslash n} + \sum_{k=1}^{K_+}\left[\mathbf{1}_{\{m_{k\backslash n}=0\}} z_{nk}\right]\right)!\right), \tag{A.29}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. As stated in Chapter 4, combining the above terms yields the following submodular objective function for $n = 1, \ldots, N$:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{Z}_{n\cdot}) = &- \frac{1}{2\sigma_{\boldsymbol{X}}^2} \boldsymbol{Z}_{n\cdot} \boldsymbol{\Phi}\boldsymbol{\Phi}^T \boldsymbol{Z}_{n\cdot}^T + \boldsymbol{Z}_{n\cdot} \boldsymbol{\omega}_{n\cdot}^T + const \\
&- \log\left(\left(K_{+\backslash n} + \sum_{k=1}^{K_+}\left[\mathbf{1}_{\{m_{k\backslash n}=0\}} z_{nk}\right]\right)!\right) \qquad \text{(A.30)} \\
\boldsymbol{\Phi}_{k\cdot} = &\left(\mathbb{E}\left[a_{k1}\right], \ldots, \mathbb{E}\left[a_{kD}\right]\right) \qquad \text{(A.31)} \\
\omega_{nk} = &\frac{1}{\sigma_{\boldsymbol{X}}^2}\left(\boldsymbol{\Phi}_{k\cdot}\boldsymbol{X}_{n\cdot}^T + \frac{1}{2}\sum_{d=1}^{D}\left[\mathbb{E}[a_{kd}]^2 - \mathbb{E}[a_{kd}^2]\right]\right) \\
&+ \nu(z_{nk}=1) - \nu(z_{nk}=0) + \mathbf{1}_{\{m_{k\backslash n}=0\}}\eta_k, \qquad \text{(A.32)}
\end{aligned}$$

$\mathbf{1}\{\cdot\}$ is the indicator function, and the subscript "$\backslash\, n$" is the value of the given variable after removing the n^{th} row from $\boldsymbol{Z}$.

A.3.5 Predictive Likelihood Estimates

For the experiments in §4.9, we estimated the predictive likelihood for held-out dimensions of a given observation. For held-out datum x'_{nd}, the predictive likelihood is:

$$p(x'_{nd}|\boldsymbol{X}) = \sum_{\boldsymbol{Z}} \int_{\boldsymbol{A}} p(x'_{nd}|\boldsymbol{Z}, \boldsymbol{A}) p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X}) \qquad \text{(A.33)}$$

where the hyperparameters are implicit.

A.3.5.1 Predictive Likelihood Estimates for Gibbs Sampling

For Gibbs samplers, we can compute unbiased estimates of the predictive likelihood as

$$p(x'_{nd}|\boldsymbol{X}) \approx \frac{1}{L}\sum_{l=1}^{L} p(x'_{nd}|\boldsymbol{Z}^l, \boldsymbol{A}^l) \qquad \text{(A.34)}$$

where l indexes the samples from each of L Gibbs sampling rounds where $\boldsymbol{Z}^l, \boldsymbol{A}^l$ are sampled from the posterior.

A.3.5.2 Predictive Likelihood Estimates for Variational Inference

For the variational methods, we use the variational distributions as proposal distributions for importance sampling from the posterior. This yields a predictive likelihood estimate similar to the Gibbs sampling estimate:

$$p(x'_{nd}|\boldsymbol{X}) = \sum_{\boldsymbol{Z}} \int_{\boldsymbol{A}} p(x'_{nd}|\boldsymbol{Z}, \boldsymbol{A}) p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X}) \tag{A.35}$$

$$\approx \sum_{l=1}^{L} p(x'_{nd}|\boldsymbol{Z}^l, \boldsymbol{A}^l) w^l \tag{A.36}$$

with $\boldsymbol{Z}^l \sim q(\boldsymbol{Z})$ and $\boldsymbol{A}^l \sim q(\boldsymbol{A})$ for $l \in [L]$, and the sample weights were

$$w^l = \frac{\frac{p(\boldsymbol{X}|\boldsymbol{Z}^l, \boldsymbol{A}^l) p([\boldsymbol{Z}^l]) p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l) q(\boldsymbol{Z}^l)}}{\sum_{l=1}^{L} \frac{p(\boldsymbol{X}|\boldsymbol{Z}^l, \boldsymbol{A}^l) p([\boldsymbol{Z}^l]) p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l) q(\boldsymbol{Z}^l)}}. \tag{A.37}$$

Unlike the Gibbs sampling estimate, however, this estimate is a ratio of two estimates and is therefore biased. Nevertheless, as discussed in §3.3.2 of Robert and Casella [2004], the bias is small and the estimator converges to the true predictive likelihood as $L \to \infty$. All estimates were computed using sampling importance resampling—see §27.6 of Barber [2012].

A.3.5.3 Predictive Likelihood Estimates for Maximization-Expectation and MAP Inference

The predictive likelihood estimate for maximization-expectation inference is the same as the predictive likelihood estimates for variational inference except the

proposal distribution $q(\boldsymbol{Z})$ was a delta function at the current MAP estimate.

$$p(x'_{nd}|\boldsymbol{X}) = \sum_{\boldsymbol{Z}} \int_{\boldsymbol{A}} p(x'_{nd}|\boldsymbol{Z}, \boldsymbol{A}) p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X}) \tag{A.38}$$

$$\approx \sum_{l=1}^{L} p(x'_{nd}|\boldsymbol{Z}^l, \boldsymbol{A}^l) w^l \tag{A.39}$$

where the sample weights were

$$w^l = \frac{\frac{p(\boldsymbol{X}|\boldsymbol{Z}^*, \boldsymbol{A}^l) p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l)}}{\sum_{l=1}^{L} \frac{p(\boldsymbol{X}|\boldsymbol{Z}^*, \boldsymbol{A}^l) p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l)}}, \tag{A.40}$$

which are the same as the weights for the variational estimate except the $\boldsymbol{Z}$ variational factors and priors cancel out when $\boldsymbol{Z}$ is fixed. This is a biased estimate for two reasons: (1) like the variational case, this estimate is a ratio of two estimates, (2) the support of $q(\boldsymbol{Z})q(\boldsymbol{A})$ does not include the support of $p(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{X})$. As previously discussed, reason (1) is not particularly bothersome, however, reason (2) can make this estimate arbitrarily biased. Because of reason (2), we also include L_2 performance on held-out dimensions as an additional evaluation criterion. The L_2 and predictive likelihood rankings were similar for all experiments and indicate that the predictive likelihood bias from the MAP estimates did not overly affect the results. Furthermore, when using the MAP $\boldsymbol{Z}$ values to initialize the sampler, the unbiased predictive likelihood from the sampler was consistent with the maximization-expectation estimates.

The predictive likelihood estimate for the fully MAP approach was the same as the maximization-expectation approach except $q(\boldsymbol{A})$ was also set to the delta function at the current MAP estimate of $\boldsymbol{A}$. In this case, the sampling weights

were

$$w^l = \frac{\frac{p(\boldsymbol{X}|\boldsymbol{Z}^l,\boldsymbol{A}^l)p([\boldsymbol{Z}^l])p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l)q(\boldsymbol{Z}^l)}}{\sum_{l=1}^{L}\frac{p(\boldsymbol{X}|\boldsymbol{Z}^l,\boldsymbol{A}^l)p([\boldsymbol{Z}^l])p(\boldsymbol{A}^l)}{q(\boldsymbol{A}^l)q(\boldsymbol{Z}^l)}} \tag{A.41}$$

$$= \frac{p(\boldsymbol{X}|\boldsymbol{Z}^*,\boldsymbol{A}^*)}{\sum_{l=1}^{L}p(\boldsymbol{X}|\boldsymbol{Z}^*,\boldsymbol{A}^*)} \tag{A.42}$$

which completely cancel and yield the following estimate of the predictive likelihood

$$p(x'_{nd}|\boldsymbol{X}) \approx p(x'_{nd}|\boldsymbol{Z}^*,\boldsymbol{A}^*). \tag{A.43}$$

As in the maximization-expectation case, this estimate can be arbitrarily biased, so we also include L_2 performance on held-out dimensions as an additional evaluation criterion.

A.4 Feige *et al.* [2011] Local Search Algorithm: Runtime Discussion

The runtime of the deterministic local search submodular maximization algorithm proposed in Feige *et al.* [2011] is $O(\frac{1}{\epsilon}K^3\log(K))$ for a ground set of size K and some parameter ϵ. We can see that this is a loose upper bound by working through the short derivation of its complexity. Specifically, the submodularity inequality states that

$$\mathcal{F}(\{w\}) - F(\emptyset) \geq \mathcal{F}(A \cup \{s\}) - \mathcal{F}(A) \tag{A.44}$$

$$\mathcal{F}(\{w\}) \geq \mathcal{F}(A \cup \{s\}) - \mathcal{F}(A) \tag{A.45}$$

for all sets A and singletons s, where w is the singleton that maximizes $\mathcal{F}$. From this inequality we have

$$K\mathcal{F}(\{w\}) \geq \mathcal{F}(A), \tag{A.46}$$

which states that $K\mathcal{F}(\{w\})$ is a global upper bound on any submodular function. The ls-algorithm only adds elements if they improve the objective function by a relative factor of $1 + \frac{\epsilon}{K^2}$, we know that after m add/remove operations to test set S we have the inequality

$$\mathcal{F}(S) \geq \left(1 + \frac{\epsilon}{K^2}\right)^m \mathcal{F}(\{w\}). \tag{A.47}$$

By transitivity we have

$$K \geq (1 + \frac{\epsilon}{K^2})^m \geq e^{m\frac{\epsilon}{K^2+\epsilon}}, \tag{A.48}$$

where the final inequality follows from the well-known log-inequality $\log(1+x) \geq \frac{1}{1+\frac{1}{x}}$. For an upper bound, we set

$$K = e^{m\frac{\epsilon}{K^2+\epsilon}}, \tag{A.49}$$

which leads to

$$m = \left(\frac{K^2}{\epsilon} + 1\right) \log(K). \tag{A.50}$$

This bounds the number of add/remove steps at

$$O(\frac{1}{\epsilon} K^2 \log(K)). \tag{A.51}$$

Each add/remove step must also find the singleton that most improves the objective function, taking $O(K)$ function queries, yielding a total $O(\frac{1}{\epsilon} K^3 \log(K))$

number of add/remove queries to the submodular function. This derivation shows that the $K^3\log(K)$ upper bound will only occur if all add/remove operations increase the objective function by a factor of exactly $(1+\frac{\epsilon}{K^2})$. If we instead require each add/remove operation to increase the objective function by at least a factor of $(1+\frac{\epsilon}{K})$, then the above derivation yields a total complexity of $O(\frac{1}{\epsilon}K^2\log(K))$.[1] The actual runtime of the ls-algorithm is problem specific, but in our use cases, the empirical complexity is *linear* for small K (roughly $K < 100$) and on the order of $K\log(K)$ for larger K.

[1] This changes the optimality guarantee from $\frac{1}{3}(1-\frac{\epsilon}{K})$OPT to $\frac{1}{3}(1-\epsilon)$OPT.

References

A. Ahmed and E. P. Xing. Seeking the truly correlated topic posterior-on tight approximate inference of logistic-normal admixture model. In *Int'l Conference on Artificial Intelligence and Statistics*, pages 19–26, 2007. 80

D. Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII1983*, pages 1–198, 1985. 3, 8

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 53

H. Attias. A variational Bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 12:209–215, 2000. 15, 16, 32, 91, 92

D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012. 96

M. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003. 16

J. M. Bernardo and A. F. Smith. *Bayesian theory*, volume 405. Wiley, 2009. 8

C. M. Bishop. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006. 15, 16

T. Broderick, M. I. Jordan, and J. Pitman. Clusters and features from combinatorial stochastic processes. *arXiv:1206.5862*, 2012. 3, 4

T. Broderick, B. Kulis, and M. I. Jordan. MAD-Bayes: MAP-based asymptotic derivations from Bayes. In *Proceedings of the 30th Int'l Conference on Machine Learning*. 2013. 52, 53, 54, 56, 80

T. Broderick, J. Pitman, and M. I. Jordan. Feature allocations, probability functions, and paintboxes. *arXiv preprint arXiv:1301.6647*, 2013. 9, 12, 14

N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *53rd Annual Symposium on Foundations of Computer Science*, pages 649–658. IEEE, 2012. iv, 37, 39, 40, 41, 81, 82

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, Series B*, 39(1):1–38, 1977. 16

N. Ding, Y. Qi, R. Xiang, I. Molloy, and N. Li. Nonparametric Bayesian matrix factorization by Power-EP. In *14th Int'l Conf. on Artificial Intelligence and Statistics*, volume 9, pages 169–176, 2010. 12, 51, 54, 69

F. Doshi-Velez and Z. Ghahramani. Accelerated sampling for the Indian buffet process. In *Proceedings of the 26th Annual Int'l Conference on Machine Learning*, pages 273–280, 2009. 52, 54, 55, 56, 60

F. Doshi-Velez, D. Knowles, S. Mohamed, and Z. Ghahramani. Large scale nonparametric Bayesian inference: Data parallelisation in the Indian buffet process. In *Advances in Neural Information Processing Systems*, volume 22, 2009. 53, 60

F. Doshi-Velez, K. T. Miller, J. Van Gael, and Y. W. Teh. Variational inference for the Indian buffet process. In *13th Int'l Conf. on Artificial Intelligence and Statistics*, pages 137–144, 2009. 4, 13, 35, 51, 54, 56, 58, 60

F. Doshi-Velez. The Indian buffet process: Scalable inference and extensions. Master's thesis, University of Cambridge, 2009. 32, 54

U. Feige, V. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011. iv, v, 37, 38, 39, 41, 48, 49, 50, 71, 98, 99, 100

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. 2

N. J. Foti and S. Williamson. A survey of non-exchangeable priors for Bayesian nonparametric models. *arXiv preprint arXiv:1211.4798*, 2012. 8

S. Gershman, M. Hoffman, and D. M. Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012. 80

Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems*, volume 13, 2001. 15, 92

Z. Ghahramani, T. L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. In *Bayesian Statistics 8*. Oxford University Press, 2007. 10, 11, 27

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. Technical report, Gatsby Unit, UCL, London, UK, 2005. 69

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems*, volume 18, 2006. i, 3, 9, 10, 27, 54

T. L. Griffiths and Z. Ghahramani. The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011. 9, 11, 14

N. L. Hjort. Nonparametric bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, pages 1259–1294, 1990. 3

A. Honkela and H. Valpola. Unsupervised variational Bayesian learning of nonlinear models. In *Advances in Neural Information Processing Systems*, pages 593–600, 2004. 80

R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *Proceedings of the 30th Annual Int'l Conference on Machine Learning*, 2013. 47, 81

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. *Independent Component Analysis and Signal Separation*, pages 381–388, 2007. 13, 73

D. Knowles and Z. Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, 5(2B):1534–1552, 2011. 13

D. Knowles and T. Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, pages 1701–1709, 2011. 80

T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *IEEE Int'l Conference on Robotics and Automation*, pages 2168 –2173, 2009. 61

V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1274–1279, 2007. 18

A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3, 2012. 19

R. Krause and D. Wild. Identifying protein complexes in high-throughput protein interaction screens using an infinite latent feature model. In *Pacific Symposium on Biocomputing*, volume 11, pages 231–242, 2006. 13

A. Krause. *Optimizing sensing: Theory and applications*. PhD thesis, Carnegie Mellon University, 2008. 19

K. Kurihara and M. Welling. Bayesian k-means as a maximization-expectation algorithm. *Neural Computation*, 21(4):1145–1172, 2008. i, 5, 16, 17, 20, 30, 70

K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005. 61

L. Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art*, pages 235–257, 1983. 17

D. Navarro and T. Griffiths. A nonparametric Bayesian method for inferring features from similarity judgments. *Advances in Neural Information Processing Systems*, 19:1033, 2007. 13

P. Orbanz and D. M. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. Obtained from `http://danroy.org/papers/OR-exchangeable.pdf`., 2013. 7, 8

K. Palla, D. Knowles, and Z. Ghahramani. An infinite latent attribute model for network data. In *Proceedings of the 29th Int'l Conference on Machine Learning*. 2012. 4, 13, 74

O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008. 67

G. Poliner and D. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2006. 61

P. Rai and H. Daume III. Beam search based map estimates for the Indian buffet process. In *Proceedings of the 28th Annual Int'l Conference on Machine Learning*, 2011. 52, 54

C. Reed and Z. Ghahramani. Scaling the Indian buffet process via submodular maximization. In *Proceedings of the 30th Int'l Conference on Machine Learning*, 2013. 5, 12

C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Citeseer, 2004. 4, 96

M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In *Int'l Conference on Independent Component Analysis and Signal Separation*, volume 5441 of *Lecture Notes in Computer Science (LNCS)*, pages 540–547. Springer, 2009. 55, 56

Y. Teh, D. Gorur, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *11th Annual Int'l Conference on Artificial Intelligence and Statistics*, 2007. 67

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In *11th Annual Int'l Conference on Artificial Intelligence and Statistics*, 2007. 3, 4, 8, 82

C. Wang and D. M. Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *Advances in Neural Information Processing Systems*, volume 25, 2012. 69

C. Wang and D. M. Blei. Variational inference in nonconjugate models. *arXiv preprint arXiv:1209.4360*, 2012. 15, 79

L. Wasserman. *All of nonparametric statistics*. Springer New York, 2006. 1

S. Williamson, C. Wang, K. Heller, and D. M. Blei. The IBP compound Dirichlet process and its application to focused topic modeling. *Proceedings of the 27th Annual Int'l Conference on Machine Learning*, 2010. 13

F. Wood and T. L. Griffiths. Particle filtering for nonparametric Bayesian matrix factorization. *Advances in Neural Information Processing Systems*, 19:1513, 2007. 53