Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 09-002

Mixed-Membership Naive Bayes Models

Hanhuai Shan and Arindam Banerjee

January 16, 2009

Mixed-Membership Naive Bayes Models

Hanhuai Shan
Dept of Computer Science & Engineering
University of Minnesota
shan@cs.umn.edu

Arindam Banerjee
Dept of Computer Science & Engineering
University of Minnesota
banerjee@cs.umn.edu

Abstract

In recent years, mixture models have found widespread usage in discovering latent cluster structure from data. A popular special case of finite mixture models are naive Bayes models, where the probability of a feature vector factorizes over the features for any given component of the mixture. Despite their popularity, naive Bayes models suffer from two important restrictions: first, they do not have a natural mechanism for handling sparsity, where each data point may have only a few observed features; and second, they do not allow objects to be generated from different latent clusters with varying degrees (i.e., mixed-memberships) in the generative process. In this paper, we first introduce marginal naive Bayes (MNB) models, which generalize naive Bayes models to handle sparsity by marginalizing over all missing features. More importantly, we propose mixed-membership naive Bayes (MMNB) models, which generalizes (marginal) naive Bayes models to allow for mixed memberships in the generative process. MMNB models can be viewed as a natural generalization of latent Dirichlet allocation (LDA) with the ability to handle heterogenous and possibly sparse feature vectors. We propose two variational inference algorithms to learn MMNB models from data. While the first exactly follows the corresponding ideas for LDA, the second uses much fewer variational parameters leading to a much faster algorithm with smaller time and space requirements. An application of the same idea in the context of topic modeling leads to a new Fast LDA algorithm. The efficacy of the proposed mixed-membership models and the fast variational inference algorithms are demonstrated by extensive experiments on a wide variety of different datasets.

1 Introduction

Probabilistic mixture models are arguably one of the most popular approaches to latent cluster structure discovery from observed data [33, 25, 5]. Naive Bayes (NB) models are a special case of such generative mixture models which have found successful applications in a wide variety of problem domains [31, 12, 30]. In NB models, the probability of a feature vector conditioned on a particular mixture component is assumed to fully factorize over individual features. In spite of their vast popularity, mixture models in general, and NB models in particular have two important restrictions that limit their modeling capabilities: first, they do not have a natural mechanism to handle sparse observations; and second, they do not allow for mixed-memberships of data points in the generative process.

Sparsity is an increasingly common property of modern large scale datasets, where data points have only a few observed features [32, 34]. For example, in a recommendation system, any user typically rates only a small fraction of all the available movies; in market-basket data, any customer

typically buys only a small fraction of all available products in a store, etc. An attempt to use standard NB models to discover user clusters based on movie ratings or customer purchase histories typically leads to unsatisfactory results, as missing entries dominate the data matrix. Since the size of modern datasets as well as their sparsity is expected to increase over the years, there is an urgent need for systematic methods for clustering of large-scale sparse datasets.

Mixture models, including NB models, assume that a data point is generated from only one of the mixture components [33, 5]. In a recommendation system scenario, such an assumption is equivalent to assuming that any user can only belong to one user cluster. If one user cluster prefers one type/genre of movies, it means that any user only likes one type/genre of movies. In reality, the assumption is clearly not true, and serves as a restriction to the modeling capability of mixture models in general, and NB models in particular. There are a few existing approaches to relax this assumption, most prominently including multi-cause models [35, 18, 37], overlapping mixture models [4, 36, 15], and aspect models [21] as well as its generalization—latent Dirichlet allocation (LDA) [7, 19]. LDA is currently a popular approach to topic modeling, where each word of a document is allowed to potentially come from a different topic, while having a fixed topic mixing proportion for each document. The topic proportions for a document is a latent variable in the model, typically with a Dirichlet prior, and serves as the mixed-membership of the document to different topics. Such mixed-membership models have advanced the state-of-the-art in topic modeling, as well as served as a basis for advanced analysis of text and relational data [7, 1]. However, such models have not been generalized to work with data which have real, categorical, or heterogenous feature vectors, and where NB models are still the method of choice [27, 41].

In this paper, we introduce a family of generative models which can handle heterogenous and sparse data observations, and allows mixed-membership clusterings, while almost maintaining the simplicity of NB models. In particular, we first introduce marginal naive Bayes (MNB) models, which generalize NB models to naturally handle sparsity in observed data by marginalizing the probability distribution over all missing features, which may be different for different data objects. Due to the conditional independence assumption over features, the marginal model is effectively a naive Bayes model over the non-missing features. For example, for the recommendation system scenario, the MNB models are defined over only the existing movie ratings, and, once learnt, can meaningfully compute probabilities of ratings which are currently unknown. More importantly, we introduce a family of mixed-membership naive Bayes (MMNB) models, effectively by taking the best of both (marginal) NB models and mixed-membership topic models such as LDA. MMNB models are significantly more flexible than (marginal) NB models, and can naturally handle sparsity. While MMNB models allow mixed-memberships of data points to all clusters, inferring the mixed-membership from observations by using expectation maximization (EM) directly is intractable. We propose two variational inference algorithms for MMNB, as well as corresponding variational EM algorithms for parameter learning for any regular exponential family distributions. The first inference algorithm is based on ideas originally proposed in the context of LDA [7]; the second algorithm uses substantially less number of variational parameters, with no dependency on the dimensionality of the dataset. An application of the same idea in the context of topic modeling gives a new Fast LDA algorithm for variational inference in LDA. By design, the new algorithm is expected to be much faster and have much smaller memory requirements.

The effectiveness of the models and ideas proposed in the paper are established through extensive experiments of various types on several datasets. A key highlight of our results is that MMNB models outperform (marginal) NB models in most settings, and the performance of MMNB is

found to be very stable across a wide range of input parameter choices, especially on held out test sets. Further, unlike (marginal) NB, MMNB generates real "soft" clusterings, demonstrating that the mixed-membership model generates qualitatively different results from simply computing the component posterior in an NB model, which tends to be close to 0 or 1. Interestingly, through properly designed experiments, we show that predictive perplexities of data points are better (lower) when the entropy of the mixed-membership is low, i.e., when the model is more sure about the data points clustering assignment. Finally, the new inference algorithm is shown to be much faster, especially for high-dimensional datasets, with no noticeable loss in accuracy. In the context of topic modeling, Fast LDA performs comparably with LDA both quantitatively and qualitatively, while being about an order of magnitude faster.

The rest of the paper is organized as follows: Section 2 gives a brief overview on generative mixture models as a background knowledge. Section 3 proposes mixed-membership naive Bayes models. We present the algorithms of variational inference in Section 4 and 5, with Section 4 giving the variational inference as a direct generalization of that in LDA, and Section 5 giving a fast variational inference. Extensive experimental results are presented in Section 6. We review the related literature in Section 7 and conclude in Section 8.

2 Generative Mixture Models

In this section, we give a brief review of the existing literature on mixture models as a background for mixture membership naive-Bayes (MMNB) models.

2.1 Finite Mixture Models

Finite mixture (FM) models are arguably the most widely studied and used form of mixture models [33, 5]. An FM model is a convex combination of a finite number of latent component distributions, each of which generates a set of observed data points. To generate each data point $\mathbf{x}$, an FM model first picks a component $z = c$ and then generates the data point following the component distribution corresponding to c. If π denotes a discrete distribution, which serves as a prior over the components, and θ_c denotes the parameters for the distribution of the c^{th} component, an FM model with k components has a density function of the following form:

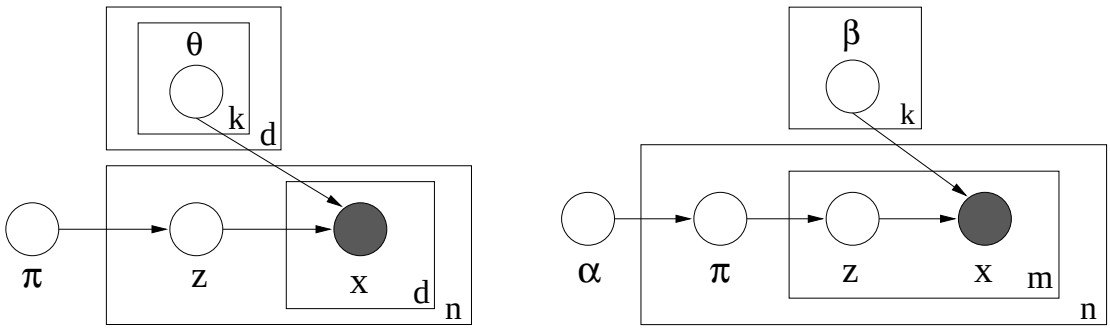
$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^{k} p(z = c|\pi_c) p(\mathbf{x}|\theta_c) \ , \tag{1}$$

where $\Theta = \{\theta_c, [c]_1^k\}$ ($[c]_1^k \equiv c = 1, \ldots, k$) are the group of parameters for the component distributions $\{p(\mathbf{x}|\theta_c), [c]_1^k\}$.

In theory, the component distributions can be from any parametric family of distributions. In practice, most of the existing literature has focussed on the case where the component distributions belong to a regular exponential family [5, 6]. Under some regularity conditions [5, 6], a regular exponential family distribution has a density function of the form

$$p_\psi(\mathbf{x}|\theta) = \exp(\langle \mathbf{x}, \theta \rangle - \psi(\theta)) p_0(\mathbf{x}) \ , \tag{2}$$

where θ is the natural parameter, $\psi(\cdot)$ is the cumulant or the log-partition function, and $p_0(\mathbf{x})$ is a non-negative base measure. Several distributions widely used for data modeling, such as Gaussian,



(a) Naive Bayes Models (b) Latent Dirichlet Allocation

Figure 1: Graphical model representation of naive Bayes models and latent Dirichlet allocation.

Bernoulli, Poisson, multinomial, gamma, beta, Dirichlet, etc., are all examples of exponential families. The choice of ψ determines a particular family, and θ determines a particular distribution in that family.

Given a set of data points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the key unsupervised learning task in an FM model is to find the "best fit" model (π^*, Θ^*). The notion of best fit is usually defined in terms of the maximum-likelihood estimate $(\pi^*, \Theta^*) = \text{argmax}_{(\pi,\Theta)}\ p(X|\pi, \Theta)$ [13, 16]. EM-style alternating maximization algorithms [33, 5] are widely used for learning such mixture models. Such algorithms alternate between computing the expectation of the likelihood (E-step) and maximizing the likelihood to obtain the parameters (M-step).

2.2 Naive Bayes Models

As a special case of FM models, naive Bayes (NB) models (Figure 1(a)) assume that features of a data point are conditionally independent given the latent component. In particular, with an appropriate univariate exponential family on feature j and component c given by

$$p_{\psi_j}(x_j|\theta_{jc}) = \exp(x_j\theta_{jc} - \psi_j(\theta_{jc}))p_j(x_j)\ ,$$

the probability of a d-dimensional feature vector $\mathbf{x}$ given the component $z = c$ is

$$p(\mathbf{x}|\theta_c) = \prod_{j=1}^{d} p_{\psi_j}(x_j|\theta_{jc}) = \prod_{j=1}^{d} \exp(x_j\theta_{jc} - \psi_j(\theta_{jc}))p_j(x_j)\ ,$$

where ψ_j determines the exponential family model appropriate for feature j, e.g., Gaussian, Poisson, etc., and θ_{jc} is the parameter corresponding to feature j and component c . Then, given the discrete distribution π over the component, the marginal probability of $\mathbf{x}$ according to the naive Bayes mixture model is given by

$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^{k} p(z = c|\pi) \prod_{j=1}^{d} p_{\psi_j}(x_j|\theta_{jc})\ . \tag{3}$$

2.3 Latent Dirichlet Allocation

One key assumption of NB models, or FM models in general, is that the latent component z is fixed across all features of a data point $\mathbf{x}$. While such an assumption is reasonable in certain domains, it puts a major restriction on the expressibility of mixture models. Latent Dirichlet allocation (LDA) [7, 19] is an elegant extension of standard mixture models by relaxing this assumption in the context of topic modeling, where each data point is a sequence of tokens, e.g., a document with a sequence of words. LDA assumes that each word in a document potentially comes from a separate topic z, which is generated from a discrete distribution π of this document, and all documents share a Dirichlet prior α. The generative process for each document $\mathbf{w}$ is as follows (Figure 1(b)):

1. Choose $\pi \sim \text{Dirichlet}(\alpha)$.
2. For each of m words (tokens) $(w_j, [j]_1^m)$ in $\mathbf{w}$:
   (a) Choose a topic (component) $z_j \sim \text{discrete}(\pi)$.
   (b) Choose w_j from $p(w_j|\beta, z_j)$.

β is a collection of parameters for k component distributions, each of which is a V dimensional discrete distribution where V is the total number of words in the dictionary.

LDA assumes that words are generated from topics, and the topics are exchangeable within a document. Recall that according to de Finetti's representation theorem [9], if the joint distribution of a set of random variables is invariant to permutation, then these random variables could be considered as independent and identically distributed conditioned on a latent parameter, which is drawn from a certain distribution. In LDA, the random variables in question are the topics corresponding to the words, and the latent parameter is the discrete distribution π, which is drawn from the Dirichlet distribution α. Then, the joint distribution of a sequence of words and their corresponding topics in a document is given by

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \int_\pi p(\pi|\alpha) \left(\prod_{j=1}^{m} p(z_j = c|\pi) p(w_j|\beta_c) \right) d\pi \ ,$$

where β_c is the parameter for the c^{th} component distribution. Marginalizing over $\mathbf{z}$ results in

$$p(\mathbf{w}|\alpha, \beta) = \int_\pi p(\pi|\alpha) \left(\prod_{j=1}^{m} \sum_{c=1}^{k} p(z_j = c|\pi) p(w_j|\beta_c) \right) d\pi \ , \tag{4}$$

Computing the probability of a collection of documents is intractable, and several approximate inference techniques have been proposed to address the problem. The two most popular approaches include variational approximation [22, 7] and Gibbs sampling [17, 19].

3 Mixed-Membership Naive Bayes Models

In this section, we first take a careful look at the strengths and limitations of naive Bayes (NB) models and latent Dirichlet allocation (LDA), and then propose mixed-membership naive Bayes models (MMNB) by taking the best of both worlds.

A "data point" for LDA [7] is a sequence of tokens, each of which is assumed to be generated from one of the discrete component distributions. The tokens are semantically identical, e.g., in case of LDA, all tokens are words. The set of distributions remains the same across all tokens. In several applications, there are two important deviations from the above set-up:

1. Each feature may have a measured value, e.g., real, categorical, etc. LDA is not designed to deal with such data since it only works with tokens.

2. Different features of a data point are semantically different. Using a homogeneous component distribution, LDA is not directly applicable to heterogenous feature vectors. By "heterogenous" feature vectors, we mean the feature vector with features of different semantics (e.g. height, weight), different data types (e.g. real, integral), different ranges of values (e.g. [-1,0], [10,100]), etc..

As for NB models, while they have been widely used due to their simplicity, and can handle heterogenous feature types with values, they also suffer from two important limitations:

1. Most large-scale datasets are sparse, so most feature values will be unknown. For example, in a movie recommendation setting, each user would have rated only a very small fraction of all available movies. NB models have no explicit mechanism to handle sparsity.

2. Unlike LDA, NB models are not mixed-membership models because they assume that all the features corresponding to a feature vector come from the same mixture component. Such a mixture of unigrams approach [7] yields simplicity, but puts a severe restriction on the modeling power of NB.

To address the first drawback of NB models, we introduce marginal naive Bayes (MNB) models by taking into consideration the sparsity structure of the data points. For a d-dimensional feature vector $\mathbf{x} = (x_1, \ldots, x_d)$ which has only a subset of m ($m \leq d$) non-missing features, following NB models, we have

$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^{k} p(z = c|\pi) \prod_{j=1}^{d} p_{\psi_j}(x_j|\theta_{jc}) \ , \tag{5}$$

where z is the latent component, π is the prior distribution over k components, $\theta_j = \{\theta_{jc}, [c]_1^k\}$ are the parameters for exponential family distributions of k components for feature j, and ψ_j determines the exponential family model appropriate for feature j. On the other hand, MNB only works with the marginal probability of the observed features. Let $\bar{\mathbf{x}} = (x_{j_1}, \ldots, x_{j_m})$ be the set of m non-missing features, where $\{j_v, [v]_1^m\}$ is a subset of $\{1, ..., d\}$. Following MNB, the marginal probability of the observed feature subset $\bar{\mathbf{x}}$ is given by

$$p(\bar{\mathbf{x}}|\pi, \Theta) = \int_{\substack{x_{j_v} \\ v \neq 1, \cdots, m}} p(\mathbf{x}|\pi, \Theta) dx_{j_v} = \sum_{c=1}^{k} p(z = c|\pi) \prod_{v=1}^{m} p_{\psi_{j_v}}(x_{j_v}|\theta_{j_v c}) \ .$$

Operationally, the model is only over the features whose values are observed, e.g., the movies that have been rated by a certain user. By abusing notation, we use $\mathbf{x}$ to denote $\bar{\mathbf{x}}$ in the sequel. Note that the observed feature sets will be potentially different for different users, and for notational

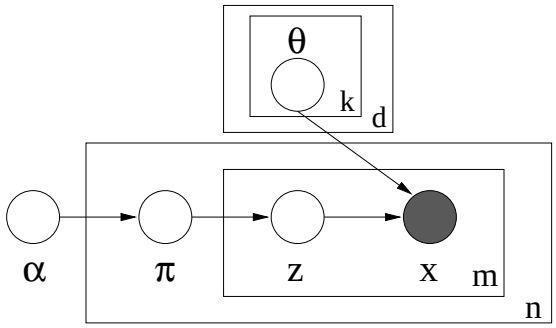


Figure 2: Graphical model representation of mixed-membership naive Bayes (MMNB) models.

convenience, we denote an observed feature j in $\mathbf{x}$ with the indicator $\exists x_j$, so that

$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^{k} p(z = c|\pi) \prod_{\substack{j=1 \\ \exists x_j}}^{d} p_{\psi_j}(x_j|\theta_{jc}) \ . \tag{6}$$

We use (6) to calculate $p(\mathbf{x}|\pi, \Theta)$ throughout this paper. Finally, note that marginalizing over missing features results in a different probability space as compared to the original NB models, so care must be taken when comparing probabilities or perplexities—we elaborate more on this issue in the context of empirical evaluations in Section 6.

By focusing only on the observed features, MNB can naturally handle sparsity, but it inherits the second problem of NB models, i.e., all features are assumed to be generated from the same component z. Meanwhile, as a mixed-membership model, LDA allows tokens in a data point to be generated from different components. We adopt the same idea in the context of MNB, and propose the *mixed-membership naive Bayes* (MMNB) models. In particular, we allow each observed feature x_j of a data point in MNB to potentially come from a separate component z_j, which is putatively generated from a latent discrete distribution π, i.e., the mixed-membership vector for that data point, with a Dirichlet prior on it. Like (marginal) naive Bayes models, the component distribution $p_{\psi_j}(x_j|\theta_{jc})$ for MMNB could be any exponential family distribution suitable for x_j, which allows MMNB to handle features with various types of measured value, so that the first limitation of LDA is conveniently addressed. In addition, as opposed to using a same β for all features in LDA, $p_{\psi_j}(x_j|\theta_{jc})$ is potentially different for different features x_j, which addresses the second limitation of LDA. Overall, as a combination of LDA and MNB, MMNB takes the best of these two to overcome the limitations of each other. However, unlike LDA, MMNB does not have the exchangeability property since if features have different semantics, permutations of features will not lead to a meaningful feature vector in that probability space.

The graphical model for MMNB is given in Figure 2. We can see that there is a Dirichlet prior α over the mixed-membership vector π for each of the n data points, and there are d sets of parameters $\Theta = \{\theta_j, [j]_1^d\}$ for d features respectively, each $\theta_j = \{\theta_{jc}, [c]_1^k\}$ containing the parameters for k components. The generative process for $\mathbf{x}$ following MMNB given α and Θ can be described as follows:

1. Choose $\pi \sim \text{Dirichlet}(\alpha)$.

2. For each non-missing feature x_j of $\mathbf{x}$:

(a) Choose a component $z_j = c \sim \text{discrete}(\pi)$.

(b) Choose a feature value $x_j \sim p_{\psi_j}(x_j|\theta_{jc})$, where ψ_j and θ_{jc} jointly decide an exponential family distribution for feature j and component c.

To make the model fully generative, we also need to generate the sparsity structure of the dataset. We can assume for the entire dataset a fixed Bernoulli distribution Bernoulli(λ), draws from which determine which features of each data point are missing. Since estimation of λ can be done from the observed sparsity structure, and, in general, it does not affect the rest of the model, we will ignore this aspect in the sequel.

From the generative model, the joint distribution of $(\pi, \mathbf{z}, \mathbf{x})$ is given by

$$p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta) = p(\pi|\alpha) \prod_{\substack{j=1 \\ \exists x_j}}^{d} p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \ . \tag{7}$$

The marginal distribution for a data point $\mathbf{x}$ is obtained by integrating over π and summing over $\mathbf{z}$. The density function for $\mathbf{x}$ with k components is given by:

$$p(\mathbf{x}|\alpha, \Theta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_j}}^{d} \sum_{c=1}^{k} p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \right) d\pi \ . \tag{8}$$

The probability of the entire dataset $\mathcal{X}$ with n data points $\mathcal{X} = \{\mathbf{x}_i, [i]_1^n\}$ is given by

$$p(\mathcal{X}|\alpha, \Theta) = \prod_{i=1}^{n} \int_{\pi} p(\pi|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_{ij}}}^{d} \sum_{c=1}^{k} p(z_{ij} = c|\pi) p_{\psi_j}(x_{ij}|\theta_{jc}) \right) d\pi \ . \tag{9}$$

In LDA, an atomic event is the generation of a token (word) w_j from a discrete component distribution over all words in the dictionary, determined by z_j. If there are k components, then there are k such discrete distributions, which are fixed for generating all words in the document. In MMNB, an atomic event is the generation of a value x_j for the j^{th} feature from an exponential family distribution $p_{\psi_j}(x_j|z_j, \theta_j)$. If there are k components and d features, the total number of component distributions would be $k \times d$, with k distributions for each feature respectively. Unlike LDA, the distribution for generating x_j not only depends on z_j, but also depends on which feature is being considered. Therefore, by choosing an appropriate exponential family distribution for each feature, MMNB is able to deal with heterogenous feature vectors. For a concrete exposition to MMNB models, we will focus on two specific instantiations of such models based on univariate Gaussian and discrete distributions for each feature in each component. Note that although the two examples we give have a same family of distributions on each of the features, MMNB allows different features to have different distributions and parameters.

MMNB-Gaussian: For each feature, MMNB-Gaussian has the same family of distributions for all features—the univariate Gaussian distribution. Note that only the distribution family is the same, but the specific distributions are still different across the features. Such models are appropriate

for the data with real-valued features. Instead of using natural parameters, we use the more common representation for Gaussian distributions parameterized by the mean μ and standard deviation σ. Assuming k latent components and d dimensional data, the model parameters are α and $\Omega = \{(\mu_{jc}, \sigma^2_{jc}), [j]_1^d, [c]_1^k\}$.[1] The probability of generating a feature vector $\mathbf{x}$ from the MMNB-Gaussian model is given by

$$p(\mathbf{x}|\alpha, \Omega) = \int_\pi p(\pi|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_j}}^{d} \sum_{c=1}^{k} p(z_j = c|\pi) p(x_j|\mu_{jc}, \sigma^2_{jc}) \right) d\pi \ . \tag{10}$$

MMNB-Discrete: Such models are appropriate for categorical features. In general, each feature is allowed to be of a different type, e.g., race, sex, etc., each potentially having a different finite number of possible values. We use the expectation parameter, i.e., the probabilities of each value's occurrence, instead of natural parameters. For each feature, the probabilities sum up to 1. In particular, assuming k latent components, d features with r_j possible values for the j^{th} feature, the model parameters are $\Omega = \{p_{jc}(r), [r]_1^{r_j}, [j]_1^d, [c]_1^k\}$ such that for latent component $z = c$ and feature j, p_{jc} is a discrete probability distribution over r_j possible values, i.e., $p_{jc}(r) \geq 0, [r]_1^{r_j}$ and $\sum_{r=1}^{r_j} p_{jc}(r) = 1$.[2] Then, the probability of generating a categorical feature vector $\mathbf{x}$ from MMNB-Discrete is given by

$$p(\mathbf{x}|\alpha, \Omega) = \int_\pi p(\pi|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_j}}^{d} \sum_{c=1}^{k} p(z_j = c|\pi) p_{jc}(x_j) \right) d\pi \tag{11}$$

4 Inference and Estimation

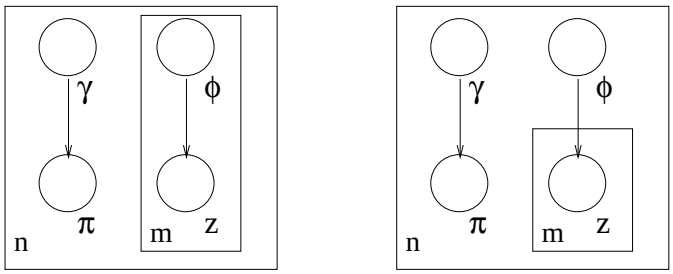
For a given dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the learning task in MMNB is to estimate the model parameters (α^*, Θ^*) such that the likelihood of observing the whole data set $p(\mathcal{X}|\alpha^*, \Theta^*)$ is maximized. A general approach for such a task is to use expectation maximization (EM) algorithms. However, the likelihood calculation in (9) is intractable, implying that a direct application of EM is not feasible. In this section, we propose a variational inference method, which alternates between obtaining a tractable lower bound to the true log-likelihood and choosing the model parameters to maximize the lower bound. To obtain a tractable lower bound, we consider an entire family of parameterized lower bounds with a set of free variational parameters, and pick the best lower bound by optimizing the lower bound with respect to the free variational parameters. For the details of derivations, please refer to Appendix A.1.

4.1 Variational Inference

In most applications of the EM algorithm for mixture modeling, in the E-step, one can directly compute the latent variable distribution [29, 3], which is used to calculate the expectation of

[1] Note that a naive Bayes model for Gaussians has the exact same set Ω of parameters.

[2] The representation is over-complete [40], and, one can use kd less parameters by using the fact that $p_{(jc)}$ is a discrete probability distribution, implying that the components will sum up to 1.



(a) MMNB/LDA (b) Fast MMNB/LDA

Figure 3: Variational distributions for MMNB/LDA and Fast MMNB/LDA.

the likelihood; in the M-step, parameter estimation is done by maximizing the expectation of the complete likelihood, where the expectation is with respect to the latent variable distribution. However, a direct computation of latent variable distribution $p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x})$ is not possible for MMNB models. In particular, the latent variable distribution, given by

$$p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x}) = \frac{p(\pi|\alpha) \prod_{j=1,\exists x_j}^{d} p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc})}{\int_\pi p(\pi|\alpha) \left(\prod_{j=1,\exists x_j}^{d} \sum_{c=1}^{k} p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \right) d\pi} \tag{12}$$

has an intractable partition, which cannot be computed in a closed form. Hence, we introduce a tractable family of parameterized distributions $q(\pi, \mathbf{z}|\gamma, \phi)$ as an approximation to $p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x})$, where (γ, ϕ) are free variational parameters. In particular, following [7], we focus on the family (Figure 3(a))

$$q(\pi, \mathbf{z}|\gamma, \phi) = q(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^{d} q(z_j|\phi_j) \ , \tag{13}$$

where for each data point, γ is a Dirichlet parameter over π and $\phi = \{\phi_j, [j]_1^d, \exists x_j\}$ are discrete distributions over the latent components z for all non-missing features. Following Jensen's inequality [29, 7] we have

$$\log p(\mathbf{x}|\alpha, \Theta) \geq E_q[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)] + H(q(\pi, \mathbf{z}|\gamma, \phi)) \ , \tag{14}$$

where $H(\cdot)$ denotes the Shannon entropy. Note that (14) gives a family of lower bounds, parameterized by (γ, ϕ), to the true likelihood $\log p(\mathbf{x}|\alpha, \Theta)$. If we denote the corresponding lower bound for data point $\mathbf{x}_i$ by $L(\gamma_i, \phi_i; \alpha, \Theta)$, following (14), we have

$$L(\gamma_i, \phi_i; \alpha, \Theta) = E_q[\log p(\pi_i|\alpha)] + E_q[\log p(\mathbf{z}_i|\pi_i)] + E_q[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] + H(q(\pi_i|\gamma_i)) + H(q(\mathbf{z}_i|\phi_i)) \ . \tag{15}$$

The lower bound of the log-likelihood on the whole dataset $\mathcal{X}$ is simply the summation of $L(\gamma_i, \phi_i; \alpha, \Theta)$ over all data points $\mathbf{x}_i$. The best lower bound can be computed by maximizing each $L(\gamma_i, \phi_i; \alpha, \Theta)$ over the free parameters (γ_i, ϕ_i). A direct calculation gives the following set of update equations

that iteratively maximize the lower bound:

$$\gamma_{ic} = \alpha_c + \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d} \phi_{ijc} \tag{16}$$

$$\phi_{ijc} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^{k}\gamma_{il}\right)\right) p_{\psi_j}(x_{ij}|\theta_{jc}) \ , \ [i]_1^n \ , \ [j]_1^d \ , \ [c]_1^k \ , \ \exists x_{ij} \ , \tag{17}$$

where γ_{ic} is the c^{th} component of the variational Dirichlet distribution for the i^{th} data point, ϕ_{ijc} is the c^{th} component of the variational discrete distribution of the j^{th} feature in the i^{th} data point, and Ψ is the digamma function, i.e., the first derivative of the log Gamma function. From [5], we know that any regular exponential family distribution $p_\psi(x|\theta) = \exp(\langle x, \theta\rangle - \psi(\theta))p_0(x)$ can be expressed in terms of the Bregman divergence between x and the expectation parameter τ as $p_\psi(x|\theta) = p_f(x|\tau) = \exp(-d_f(x,\tau))b_f(x)$, where $b_f = \exp(f(x))p_0(x)$, and $d_f(\cdot,\cdot)$ is the Bregman divergence determined by the function f, which is the conjugate of the cumulant function ψ of the family. Therefore, (17) could be written as

$$\phi_{ijc} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^{k}\gamma_{il}\right) - d_{f_j}(x_{ij}, \tau_{jc})\right) \ , \tag{18}$$

where τ_{jc} is the mean of the j^{th} feature of the c^{th} component. The above equation shows that ϕ_{ijc} is inversely proportional to the exponential of Bregman divergence between the j^{th} feature and its expectation of the c^{th} component, i.e., if x_{ij} is far from the mean τ_{jc}, its membership in component c will be small. In fact, $\phi_{ij} = \{\phi_{ijc}, [c]_1^k\}$ gives the mixed-membership of the j^{th} feature belonging to k components respectively. For a specific model, such as MMNB-Gaussian, the updating equation for ϕ_{ijc} could be obtained by replacing the corresponding distributions in place of $p_{\psi_j}(x_{ij}|\theta_{jc})$ in (17). The form of the updates for γ_{ic} is independent of the exponential family being used.

4.2 Parameter Estimation

The goal of parameter estimation is to obtain (α, Θ) such that $\log p(\mathcal{X}|\alpha, \Theta)$ is maximized. Since the log-likelihood is intractable, we use the lower bound as a surrogate objective to be maximized. Note that for a fixed value of the variational parameters (γ_i^*, ϕ_i^*) obtained by variational inference for each $\mathbf{x}_i$, the lower bound of $\log p(\mathcal{X}|\alpha, \Theta)$, $\sum_{i=1}^n L(\gamma_i^*, \phi_i^*; \alpha, \Theta)$, is a function of the parameters (α, Θ). Following [33, 5], the parameters Θ can be estimated in a closed form for all exponential family distributions.

From the Bregman divergence perspective, let τ_{jc} be the expectation parameter for the j^{th} feature of the c^{th} component, the estimation for τ_{jc} is given by

$$\tau_{jc} = \frac{\sum_{i=1,\exists x_{ij}}^n \phi_{ijc} s_{ij}}{\sum_{i=1,\exists x_{ij}}^n \phi_{ijc}} \ , \ [j]_1^d \ , \ [c]_1^k \ , \tag{19}$$

where s_{ij} is the sufficient statistic and the natural parameter θ_{jc} is given by conjugacy as

$$\theta_{jc} = \nabla f_j(\tau_{jc}) \ , \ [j]_1^d \ , \ [c]_1^k \ ,$$

where $f_j(\cdot)$ is the conjugate of cumulant function ψ_j for each feature. We now give the parameter estimation for two special cases—MMNB-Gaussian and MMNB-Discrete.

MMNB-Gaussian: For Gaussians, by maximizing the lower bound, the exact update equations for μ_{jc} and σ_{jc} can be obtained as

$$\mu_{jc} = \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ijc} x_{ij}}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ijc}} \tag{20}$$

$$\sigma_{jc}^2 = \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ijc}(x_{ij} - \mu_{jc})^2}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ijc}} \ , \ [j]_1^d \ , \ [c]_1^k \ . \tag{21}$$

MMNB-Discrete: For a discrete distribution p_{jc} over $r = 1, \ldots, r_j$ values for feature j, the estimate of $p_{jc}(r)$ is given by

$$p_{jc}(r) = \sum_{i=1}^{n} \phi_{ijc} \mathbb{1}(x_{ij} = r) \ , \ [c]_1^k \ , \ [j]_1^d \ , \ [r]_1^{r_j} \ , \tag{22}$$

where $\mathbb{1}(x_{ij} = r)$ is the indicator of observing value r for feature j in observation $\mathbf{x}_i$. While such a maximum likelihood (ML) estimate will give the maximizing parameters on an observed training set, there is a possibility of some probability estimates being zero. Such an eventuality does not pose a problem on the training set, but inference on unseen or testing data may become problematic. If a feature in the test set takes a value that it has not taken in the entire training set, the model will assign a zero probability to the entire set of testing observations. The standard approach to address the problem is to use smoothing, so that none of the estimated parameters is zero. In particular, we use Laplace smoothing, which results from a maximum a posteriori (MAP) estimate [10] assuming a Dirichlet prior over each discrete distribution, so that

$$p_{jc}(r) = \sum_{i=1}^{n} \phi_{ijc} \mathbb{1}(x_{ij} = r) + \epsilon \ , \ [c]_1^k \ , \ [j]_1^d \ , \ [r]_1^{r_j} \ , \tag{23}$$

for some $\epsilon > 0$.

The update of α is independent of the choice of exponential family distribution. Using Newton-Raphson algorithm [7, 26] with line search, the updating equation is given by:

$$\alpha_c' = \alpha_c - \eta \frac{g_c - u}{h_c} \ , \ [c]_1^k \ , \tag{24}$$

where

$$\begin{aligned} g_c &= n\left(\Psi\left(\sum_{l=1}^{k} \alpha_l\right) - \Psi(\alpha_c)\right) + \sum_{i=1}^{n}\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{c=1}^{k} \gamma_{ic}\right)\right) \\ h_c &= -n\Psi'(\alpha_c) \\ u &= \frac{\sum_{c=1}^{k} g_c/h_c}{w^{-1} + \sum_{c=1}^{k} h_c^{-1}} \\ w &= n\Psi'(\sum_{c=1}^{k} \alpha_c) \ . \end{aligned}$$

Since α has the constraint of $\alpha_c > 0$, by multiplying the second term of (24) by η, we are performing a line search to prevent α_c to go out of the feasible range. At the beginning of each iteration, we set η to be 1. If the updated α_c falls into the feasible range, the algorithm goes on to the next iteration, otherwise, it reduces α by a factor of 0.5 until the updated α_c becomes valid.

4.3 EM for MMNB

Based on the variational inference and parameter estimation updates, it is straightforward to construct an EM algorithm to estimate (α, Θ). Starting with an initial guess (α_0, Θ_0), the EM algorithm alternates between two steps:

1. E-Step: Given $(\alpha^{(t-1)}, \Theta^{(t-1)})$, for each data point $\mathbf{x}_i$, find the optimal variational parameters

$$(\gamma_i^{(t)}, \phi_i^{(t)}) = \underset{(\gamma_i, \phi_i)}{\operatorname{argmax}}\ L(\gamma_i, \phi_i; \alpha^{(t-1)}, \Theta^{(t-1)}) \ .$$

   $L(\gamma_i^{(t)}, \phi_i^{(t)}; \alpha, \Theta)$ gives a lower bound to $\log p(\mathbf{x}_i|\alpha, \Theta)$.

2. M-Step: An improved estimate of model parameters (α, Θ) are obtained by maximizing the aggregate lower bound:

$$(\alpha^{(t)}, \Theta^{(t)}) = \underset{(\alpha, \Theta)}{\operatorname{argmax}}\ \sum_{i=1}^{n} L(\gamma_i^{(t)}, \phi_i^{(t)}; \alpha, \Theta) \ .$$

After t iterations, the objective function becomes $L(\gamma_i^{(t)}, \phi_i^{(t)}; \alpha^{(t)}, \Theta^{(t)})$. In the $(t+1)^{th}$ iteration, we have

$$\sum_{i=1}^{n} L(\gamma_i^{(t)}, \phi_i^{(t)}; \alpha^{(t)}, \Theta^{(t)}) \leq \sum_{i=1}^{n} L(\gamma_i^{(t+1)}, \phi_i^{(t+1)}; \alpha^{(t)}, \Theta^{(t)}) \leq \sum_{i=1}^{n} L(\gamma_i^{(t+1)}, \phi_i^{(t+1)}; \alpha^{(t+1)}, \Theta^{(t+1)}) \ .$$

The first inequality holds because in the E-step, $(\gamma_i^{(t+1)}, \phi_i^{(t+1)})$ maximizes $L(\gamma_i, \phi_i; \alpha^{(t)}, \Theta^{(t)})$. The second inequality holds because in the M-step, $(\alpha^{(t+1)}, \Theta^{(t+1)})$ maximizes $(\gamma_i^{(t+1)}, \phi_i^{(t+1)}; \alpha, \Theta)$. Therefore, the objective function is non-decreasing until convergence.

5 Fast Variational Inference

The variational distribution we have introduced in Section 4 exactly follows the idea proposed for latent Dirichlet allocation (LDA) [7], where every feature j of the data point $\mathbf{x}_i$ has a corresponding variational parameter ϕ_{ij} for the discrete distribution. In this section, we introduce a different variational distribution with a smaller number of parameters, yielding a much faster variational inference, which we call Fast MMNB. We also extend the idea to LDA and come up with the Fast LDA algorithm. The details of derivation are presented in Appendix A.2.

5.1 Variational Approximation

Given the lower bound for log-likelihood of each data point as (14) in Section 3, the variational distribution we have used is (13), where each non-missing feature j of each data point $\mathbf{x}_i$ has a separate discrete distribution ϕ_{ij}. In a full data matrix with n d-dimensional data points, the total number of ϕ_{ij} would be $n \times d$, which is a huge number for high-dimensional data. Meanwhile, since in the E-step of EM algorithm, the optimization is done over each variational parameter, a large number of variational parameters will lead to a large number of optimizations, significantly slowing the algorithm down. To make the algorithm more efficient, we introduce a new family of variational distributions (Figure 3(b)):

$$q'(\pi, \mathbf{z}|\phi, \gamma) = q'(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^{d} q'(z_j|\phi) \ . \tag{25}$$

Compared with $q(\pi, \mathbf{z}|\phi, \gamma)$ in (13), $q'(\pi, \mathbf{z}|\phi, \gamma)$ only has one discrete distribution parameter ϕ over all latent components z for features of each data point, making $q'(\pi, \mathbf{z}|\phi, \gamma)$ closer to the original model, which also has only one discrete distribution π for each data point. In comparison, $q(\pi, \mathbf{z}|\phi, \gamma)$ is substantially over parameterized having a ϕ_j for each feature j. If there are totally n data points, the total number of ϕs decreases from $n \times d$ in (13) to n in (25), accordingly, the number of optimizations over ϕ also decreases from $n \times d$ to n. Such a reduction would imply a big saving on both time and space, especially for high dimensional data with a large d.

Assuming there are m_i non-missing features for each data point $\mathbf{x}_i$. Given the variational distribution in (25), we have a set of new lower bounds $L(\gamma_i, \phi_i; \alpha, \Theta)$ for $p(\mathbf{x}_i|\alpha, \Theta)$, and the best lower bound is obtained by maximizing $L(\gamma_i, \phi_i; \alpha, \Theta)$ with respect to the variational parameters. The update equations for variational parameters become

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \tag{26}$$

$$\phi_{ic} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^{k} \gamma_{il}\right)\right) \left(\prod_{\substack{j=1 \\ \exists x_{ij}}}^{d} p_{\psi_j}(x_{ij}|\theta_{jc})\right)^{1/m_i} , \ [i]_1^k \ , \ [c]_1^k \ , \tag{27}$$

where γ_{ic} and ϕ_{ic} are the variational Dirichlet distribution and discrete distribution for the c^{th} component of $\mathbf{x}_i$ respectively. Comparing (27) to (17), we can see that instead of having the term $p_{\psi_j}(x_{ij}|\theta_{jc})$ in ϕ_{ijc} for each feature j of $\mathbf{x}_i$, since there is only one ϕ_i for all features of $\mathbf{x}_i$, (27) takes the geometric mean of $p_{\psi_j}(x_{ij}|\theta_{jc})$ over all non-missing features of $\mathbf{x}_i$. γ_{ic} is again independent of the exponential family being used.

5.2 Parameter Estimation

After obtaining the variational parameters, we can obtain a tractable lower bound of the log-likelihood as a function of the model parameters (α, Θ). The estimation for α is the same as in Section 4 using Newton-Raphson algorithm with line search, and the estimation for Θ has a closed form for exponential family distributions. We show the expressions for Gaussian and discrete cases. From the Bregman divergence perspective, assuming the expectation parameter for the j^{th} feature

of component c is τ_{jc}, the estimation for τ_{jc} is given by

$$\tau_{jc} = \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic} s_{ij}}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}} \ , \ [j]_1^d \ , \ [c]_1^k \ , \tag{28}$$

where s_{ij} is the sufficient statistic and the natural parameter $\theta_{jc} = \nabla f_j(\tau_{jc})$ by conjugacy, where $f_j(\cdot)$ is the conjugate of cumulant function ψ_j for each feature. For two special cases—MMNB-Gaussian and MMNB-Discrete, the closed form parameter estimates are given below. Note that (28)-(31) are mild variants of (19)-(22) as ϕ_{ic} does not depend on feature j.

MMNB-Gaussian: For Gaussians, the update equations for μ_{jc} and σ_{jc}^2 are given by

$$\mu_{jc} = \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic} x_{ij}}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}} \tag{29}$$

$$\sigma_{jc}^2 = \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic} (x_{ij} - \mu_{jc})^2}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}} \ , \ [c]_1^k \ , \ [j]_1^d \ . \tag{30}$$

MMNB-Discrete: For a discrete distribution p_{jc} over $r = 1, \ldots, r_j$ values for feature j, the update equation for $p_{jc}(r)$ is given by

$$p_{jc}(r) = \sum_{i=1}^{n} \phi_{jc} \mathbb{1}(x_{ij} = r) + \epsilon \ , \ [c]_1^k \ , \ [j]_1^d \ , \ [r]_1^{r_j} \tag{31}$$

where $\mathbb{1}(x_{ij} = r)$ is the indicator of observing value r for feature j in observation $\mathbf{x}_i$.

Given the updates for variational and model parameters, an EM algorithm could be constructed to estimate (α, Θ) as in Section 4.3.

5.3 Fast LDA

We apply the same idea to variational inference in LDA [7] to obtain Fast LDA. As in Figure 1(b), LDA has two model parameters α and β: α is the parameter of the Dirichlet distribution over π, and β is the set of the discrete distribution parameters for each of k components over V words, where V is the size of the dictionary. Following the notation in LDA [7], the v^{th} word in the dictionary is represented by a V-dimensional vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$, and each document $\mathbf{w}$ is represented by m words $\mathbf{w} = \{w_1, w_2, ..., w_m\}$.

We introduce the same variational distribution as in Figure 3(b), i.e., for each document $\mathbf{w}$, we introduce one Dirichlet distribution parameterized by γ and one discrete distribution parameterized by ϕ. In particular, the variational distribution is given by:

$$q(\pi, \mathbf{z}|\phi, \gamma) = q(\pi|\gamma) \prod_{j=1}^{m} q(z_j|\phi) \ . \tag{32}$$

The lower bound of the log-likelihood in (4) is again obtained from Jensen's inequality as in (14). By taking derivative of the lower bound with respect to ϕ and γ respectively and setting them to

zero, the update equations for variational parameters of $\mathbf{w}_i$ is as follows:

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \tag{33}$$

$$\phi_{ic} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^{k} \gamma_{il} \right) + \frac{1}{m_i} \sum_{j=1}^{m_i} \sum_{v=1}^{V} w_{ij}^{v} \log \beta_{cv} \right) \ , \ [i]_1^n \ , \ [c]_1^k \ , \tag{34}$$

$$\tag{35}$$

where m_i is the number of words in document $\mathbf{w}_i$.

For fixed values of variational parameters γ and ϕ, maximizing the aggregate lower bound with respect to the model parameters yields the solution for α and β. In particular, the solution for α is the same as (24), and the update equation for β is given by:

$$\beta_{cv} \propto \sum_{i=1}^{n} \left(\phi_{ic} \sum_{j=1}^{m_i} w_{ij}^{v} \right) \ , \ [c]_1^k \ , \ [v]_1^V \ . \tag{36}$$

6 Experimental Results

In this section, we present three sets of experimental results to assess the performance achieved by MMNB: (1) comparing MMNB with MNB in terms of clustering accuracy and modeling performance, (2) comparing MMNB with LDA in terms of modeling and prediction performance, and (3) comparing Fast LDA with LDA in terms of topic modeling performance and processing speed.

6.1 Datasets

Various datasets with different data types (real, integral, discrete, etc.) and different sparsity structures (full, sparse) are used in our experiments to show the versatility of the proposed family of algorithms.

UCI Datasets: Nine datasets from UCI machine learning repository are used for our experiments. These datasets are represented as real-valued full matrices without missing entries. The numbers of instances, features and classes in each dataset are listed in Table 1.

Movielens: Movielens is a movie recommendation dataset created by the Grouplens Research Project.[3] It contains 100,000 ratings for 1682 movies by 943 users represented as a sparse matrix, i.e., there are only 6.30% non-missing entries in the matrix. The ratings range from 1 to 5 with 5 being the best. We binarize the dataset such that entries with rating 4 or 5 become 1 and other non-missing entries become 0. We use both the original data as well as the binarized data in the experiments.

Foodmart: Foodmart data comes with Microsoft SQL server. It contains transaction data for a fictitious retailer. In particular, there are 164,558 sales records for 7803 customers and 1559 products, i.e., there are only 1.35% non-missing entries in the matrix. Each customer record contains the number of each product bought by the customer. We binarize the dataset such that entries

[3] http://www.grouplens.org/node/73

Dataset	Instances	Features	Classes
Ecoli	336	7	8
Glass	214	9	6
Ionosphere	351	32	2
Statlog-seg	2310	19	7
Segmentation	210	19	7
Sonar	208	60	2
Vowel	990	11	11
Wdbc	569	30	2
Wine	178	13	3

Table 1: The number of instances, features and classes in each UCI dataset.

above the median of all valid entries become 1 and other non-missing entries become 0. Further, we remove rows and columns with less than 10 non-missing entries. We use both the original data as well as the binarized data in the experiments.

Jester: Jester is a joke rating dataset.[4] The original dataset contains 4.1 million continuous ratings of 100 jokes from 73,421 users. The ratings are ranged from -10 to 10 with 10 the best. We pick 1000 users who rate all 100 jokes and use this dense data matrix in our experiment. We again binarize the dataset such that the non-negative entries become 1 and the negative entries become 0. We use both the original data as well as the binarized data in the experiments.

For comparing Fast LDA with LDA, we use 3 text datasets:

NASA: NASA is a text dataset downloaded from Aviation Safety Reporting System (ASRS) online database.[5] This database contains narratives submitted by pilots to report problems in flights. The dataset used is a subset of the whole database. It contains 4226 documents originated by three sources: flight crew, maintenance, and passengers. Each document is represented as a word vector of length 604, where each entry is the number of times a word appears in the document.

Classic3: Classic3 [11] is a well known text dataset. It contains 3893 documents from three different classes including aeronautics, medicine and information retrieval. Each document is represented as a word vector of length 5923, where each entry is the number of times a word appears in the document.

CMU Newsgroup: The CMU Newsgroup is also a benchmark text dataset [24]. The standard dataset of CMU Newsgroup contains 19,997 messages, collected from 20 different USENET newsgroups. We use two subsets in our experiments: (1) **CMU-Diff** is a collection of 3000 messages from 3 different newsgroups with 1000 messages each topic: alt.atheism, rec.sport.baseball and sci.space. The dimension for the vector of each document is 7666. (2) **CMU-Sim** is a collection of 3000 messages from 3 different newsgroups with 1000 messages each topic: talk.politics.guns, talk.politics.mideast, talk.politics.misc. The dimension of each document is 10083. Each entry is

[4] http://goldberg.berkeley.edu/jester-data/

[5] http://akama.arc.nasa.gov/ASRSDBOnline/QueryWizard_Begin.aspx

the number of times a word appears in the document.

6.2 Methodology

We use three criteria for evaluation: perplexity, micro-precision, and mutual information with a train-test split and 10-fold cross validation as described below. Among the three criteria, micro-precision and mutual-information need class labels, but perplexity does not.

Perplexity: Both MMNB and MNB are capable of assigning a log-likelihood $\log p(\mathbf{x}_i)$ to each observed data point $\mathbf{x}_i$. Based on the log-likelihood scores, we compute the perplexity [21, 7] of the entire dataset $\mathcal{X}$ as

$$Perplexity(\mathcal{X}) = \exp\Big\{ -\frac{\sum_{i=1}^{n} \log p(\mathbf{x}_i)}{\sum_{i=1}^{n} m_i} \Big\} , \tag{37}$$

where m_i is the number of observed features for $\mathbf{x}_i$ and n is the number of data points. In the case of a full matrix such as the UCI data, m_i is the number of features, which is the same for all data points. In the case of a sparse matrix such as Movielens, m_i may be different for different data points. As shown in (37), the perplexity is a monotonically decreasing function of the log-likelihood, implying that *lower perplexity is better* (especially on the test set) since the model can explain the data better.

Micro-precision: We use micro-precision [28] to evaluate the accuracy of clustering. MMNB and MNB generate the posterior probability of each data point belonging to k latent clusters (soft clustering). We pick the cluster with the highest probability as the predicted cluster (hard clustering). Each predicted cluster is then mapped to the true class which has the most overlapping data points with the cluster among all true classes. Therefore the mapping between the predicted cluster and the true class is "many to one", i.e., more than one predicted clusters could be mapped to a same true class, but each predicted cluster could only be mapped to one true class. For each predicted cluster c, we define "correctly clustered" data points as the overlapping data points with the corresponding true class. Denoting the number of correctly clustered data points for each cluster c with n_c, the micro-precision could be defined as [28]:

$$MP = \frac{\sum_{c=1}^{k} n_c}{n} , \tag{38}$$

where k is the total number of clusters, and n is the total number of data points.

Mutual Information: Given the hard clusterings, we also calculate mutual information [8, 39] which evaluates the amount of statistical similarity between the clusters and true classes. If Z is a random variable for the cluster assignments and Y is a random variable for the true classes on the same data, then their mutual information is given by

$$I(Z;Y) = \sum_{Z}\sum_{Y} p(Z,Y) \log \frac{p(Z,Y)}{p(Z)p(Y)} , \tag{39}$$

where $p(Z,Y)$ is the joint distribution of Z and Y; $p(Z)$ and $p(Y)$ are marginal distributions for Z and Y respectively.

	Training Set		Test set	
	MMNB	NB	MMNB	NB
Ecoli	0.0120±0.00037	**0.0105±0.0002**	0.0134±0.0018	**0.0115±0.0001**
Glass	**0.0649±0.0062**	0.1800±0.0316	**0.0783±0.0232**	0.2383±0.1415
Ionosphere	**1.4241±0.0948**	1.6558±0.0228	**1.5035±0.1878**	1.7093±0.1863
Statlog-seg	**1.3013±0.0681**	2.1654±0.2250	**1.3032±0.0517**	2.2422±0.2536
Segmentation	**1.1991±0.1062**	1.8365±0.1649	**1.4215±0.1770**	2.6365±1.3470
Sonar	**0.2934±0.0044**	0.3060±0.0024	**0.3043±0.0147**	0.3161±0.0146
Vowel	0.7190±0.0226	**0.6063±0.0105**	0.7709±0.0220	**0.6416±0.0209**
Wdbc	**0.7797±0.0151**	0.7862±0.0107	**0.7974±0.0784**	0.8090±0.0874
Wine	**4.2216±0.2329**	4.5212±0.1415	**4.5722±0.6454**	5.0251±0.4230

Table 2: Perplexity for MMNB and NB on UCI datasets. MMNB has a lower perplexity on most of the datasets.

	Training Set		Test set	
	MMNB	NB	MMNB	NB
Ecoli	0.7468±0.0147	**0.7764±0.0205**	**0.8334±0.1127**	0.7656±0.1009
Glass	**0.5450±0.0501**	0.5095±0.0399	**0.6389±0.0744**	0.5976±0.0748
Ionosphere	0.6422±0.0069	**0.7057±0.0174**	0.6514±0.0500	**0.6886±0.0767**
Statlog-seg	**0.5675±0.0516**	0.5383±0.0465	**0.5874±0.0544**	0.5619±0.0675
Segmentation	**0.5783±0.0396**	0.5037±0.0530	**0.6476±0.0930**	0.6333±0.1100
Sonar	**0.5706±0.0265**	0.5661±0.0100	**0.6051±0.0550**	0.6050±0.0438
Vowel	**0.3249±0.0273**	0.2470±0.0368	**0.3990±0.0258**	0.3222±0.0450)
Wdbc	**0.9214±0.0109**	0.9131±0.0046	**0.9161±0.0253**	0.9089±0.0351
Wine	**0.9235±0.0795**	0.6823±0.0213	**0.9294±0.0668**	0.6882±0.0834

Table 3: Micro-precision for MMNB and NB on UCI datasets. MMNB has a higher micro-precision on most of the datasets.

Unless otherwise specified, we use 10-fold cross-validation with random initializations. In a 10-fold cross-validation, we divide the dataset evenly into 10 parts, one of which is picked as the test set, and the remaining 9 parts are used as the training set. The process is repeated for 10 times, with each part used exactly once as the test set. We then take the average of results over 10 folds on training set and test set respectively. For results on the training set, we train the model on training data by running EM as in Section 4.3 until convergence to obtain the model parameters and variational parameters, which are used to calculate the perplexity, micro-precision and mutual information. For results on test sets, given the model parameters from the training process, we run E-step (inference) on test data to obtain the variational parameters, then perplexity, micro-precision and mutual information are calculated.

6.3 MMNB vs. MNB

In this section, we demonstrate the efficacy of MMNB through the comparison with MNB (or NB for the datasets without missing entries, such as UCI data) on several datasets with different data types. For each dataset, we choose an appropriate distribution depending on the feature type as the generative model. The results show that MMNB is applicable to different types of data and it achieves a much better performance than MNB or NB.

	Training Set		Test set	
	MMNB	NB	MMNB	NB
Ecoli	0.8811±0.0210	**0.9684±0.0342**	**0.8157±0.3124**	0.7887±0.2370
Glass	**0.4931±0.0772**	0.4896±0.0965	**0.5999±0.2538**	0.5655±0.2446
Ionosphere	0.0762±0.0231	**0.1524±0.0159**	0.0785±0.0459	**0.1569±0.0790**
Statlog-seg	**1.0133±0.1082**	0.9878±0.1042	**1.0373±0.1018**	1.0254±0.1351
Segmentation	**1.0383±0.0847**	0.8669±0.1517	**1.1159±0.2078**	0.9892±0.2653
Sonar	**0.0166±0.0127**	0.0100±0.0038	**0.0306±0.0327**	0.0239±0.0204
Vowel	**0.7386±0.0736**	0.5636±0.1389	**0.8970±0.0677**	0.6507±0.1600
Wdbc	**0.3856±0.0288**	0.3731±0.0128	**0.3813±0.0901**	0.3734±0.0880
Wine	**0.8680±0.1070**	0.4990±0.0220	**0.8727±0.1139**	0.4563±0.1234

Table 4: Mutual information for MMNB and NB on UCI datasets. MMNB has a higher mutual information on most of the datasets.

6.3.1 Methodology

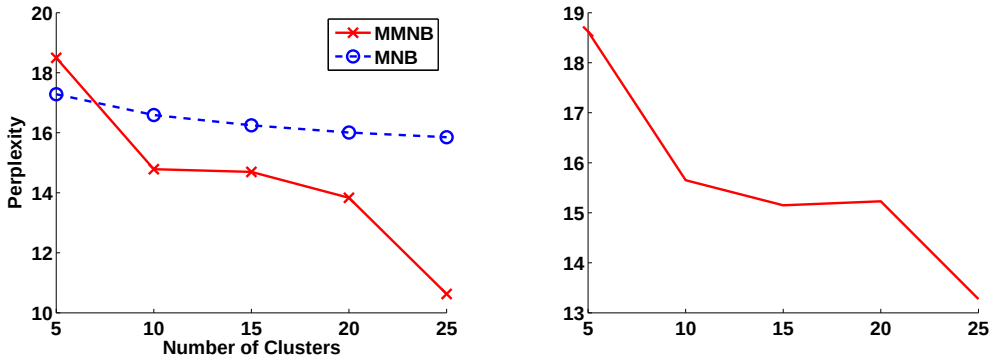
We run experiments on UCI datasets, as well as on Jester, Foodmart and Movielens using MNB and MMNB respectively. Note that for the matrix without missng entries such as UCI data, MNB is equivalent to NB, so we may use "MNB" and "NB" interchangeably in the sequel. For UCI and Jester, we use Gaussian distribution as the generative model; for Foodmart, we use Poisson[6]; and for Movielens, we use discrete distribution. The number of clusters we use for UCI data is the actual number of classes given in the dataset, and we try different number of classes for Jester, Movielens and Foodmart respectively.

Before we make the comparison between MMNB and NB, we must note that NB effectively has one less of freedom in parameter than MMNB. In particular, the Dirichlet distribution α in MMNB can be any non-negative vector, whereas the discrete distribution π in NB has to be a probability distribution summing up to one. In other words, if there are k scalers to determine parameter α, there will be only $k-1$ scalers to determine the parameter π. For a generative model, a larger number of parameters may yield a better performance on the training set, such as a lower perplexity or a higher accuracy, since the model could be as complicated as necessary to fit the training data perfectly well. However, such complicated models typically loses the ability for generalization and leads to over-fitting on test set. In our experiments, however, we consider the comparison to be fair due to the following two reasons: First, MMNB and NB essentially have the same number of parameters, with NB having one less degree of freedom on the prior parameter. Second, we compare the performance on both training and test sets. If the over-fitting does occur to MMNB, it will result in a bad performance on test set. Thus the results on test sets are more interesting and crucial.

6.3.2 Results

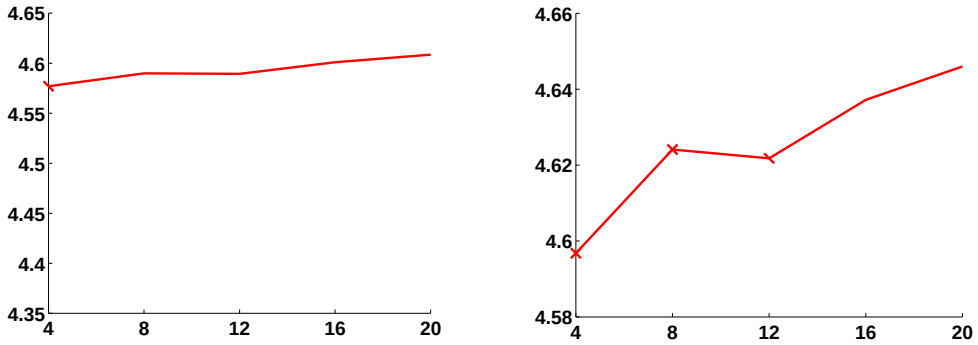
In this section, we present two parts of results. The first part is the comparison between MMNB and NB in terms of perplexity, micro-precision and mutual information. The second part is some interesting result demonstrating the property of MMNB's behavior.

[6]For Foodmart data, there is one unit right shift of Poisson distribution since the value of non-missing entries starts from 1 instead of 0, so we subtract 1 from all non-missing entries to shift the distribution back.



(a) Training set perplexity (b) Test set perplexity

Figure 4: Perplexities of MMNB and MNB with various number of clusters on Jester.



(a) Training set perplexity (b) Test set perplexity

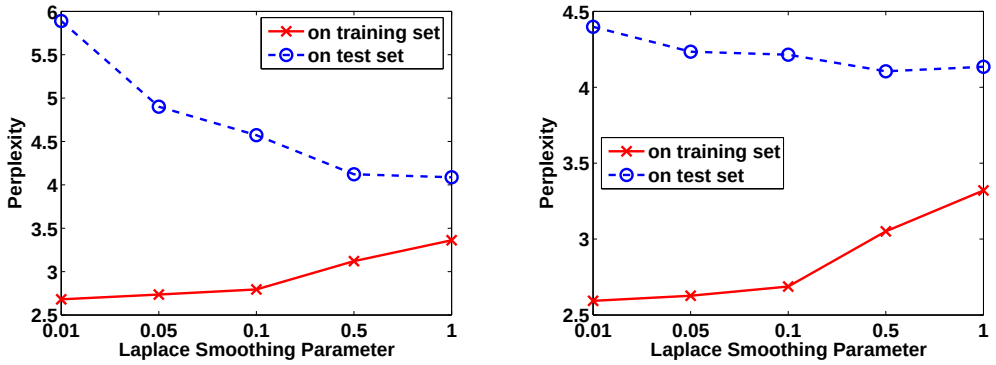
Figure 5: Perplexities of MMNB and MNB with various number of clusters on Foodmart.

MMNB-NB Comparison

The average perplexity, micro-precision, and mutual information of MMNB and NB on UCI data after a 10-fold cross-validation are listed in Table 2-Table 4 respectively. It is clear that MMNB has a lower perplexity than NB on most datasets, indicating that MMNB fits the data better than NB. In terms of micro-precision and mutual information, MMNB also wins most of the times, especially on test sets, which is a convincing evidence for MMNB's higher performance in clustering.

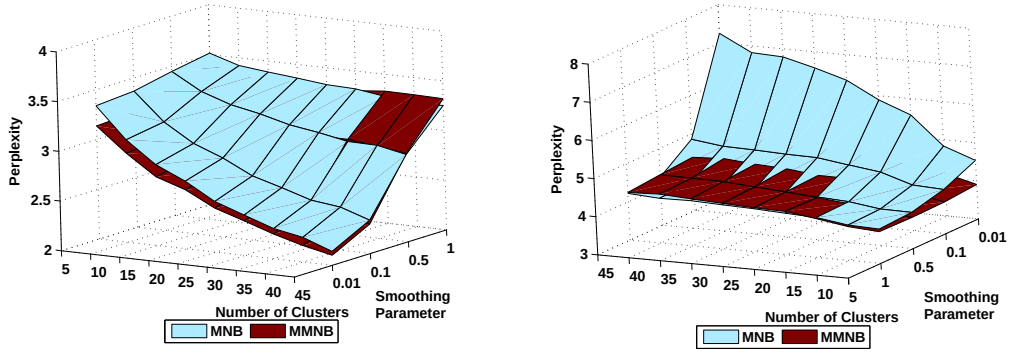
We also compare the perplexity of MMNB and MNB on Jester, Movielens, and Foodmart. Compared to the UCI data, these three datasets are closer to data in real applications because of the larger number of data points and higher dimensions of feature vectors. Note that since these three datasets do not have true class labels, we cannot compare the micro-precision and mutual information on them. The perplexities on Jester and Foodmart are presented in Figure 4 and 5. The number of clusters for Jester is varied from 5 to 25 in steps of 5 and that for Foodmart is varied from 4 to 20 in steps of 4. On Jester, MMNB mostly outperforms MNB with varying number of classes on both training and test sets. On Foodmart, although MNB achieves a lower perplexity on training set, it indicates over-fitting, especially with larger number of clusters, since its corresponding perplexity on test set is very high. In comparison, MMNB is more robust.

We perform more detailed experiments on Movielens. Given a fixed number of classes (k=20),



(a) MNB (b) MMNB

Figure 6: Perplexities of MNB and MMNB with $k = 20$ and varying ϵ on Movielens. Perplexity decreases with larger smoothing parameter on training set, and increases on test set.



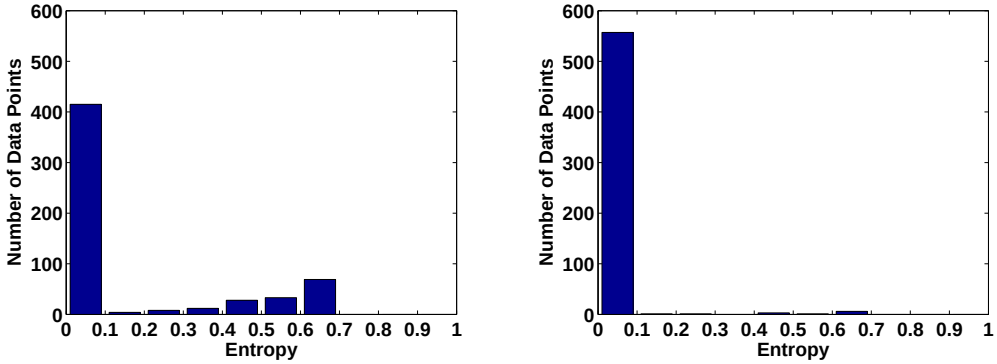
(a) Training set (b) Test set

Figure 7: Perplexity surfaces of MNB and MMNB over a range of k and ϵ on Movielens. MMNB mostly has a lower perplexity than MNB, and a more stable performance on test set.

Figure 6 reports the perplexities of MMNB and MNB with ϵ varied from 0.01 to 1, where ϵ is the Laplace smoothing parameter as introduced in Section 4.2 for MMNB-Discrete case. The overall trend is as follows: when ϵ increases, the perplexity on training set increases and the perplexity on test set decreases. The result is consistent with the Bayesian intuition behind smoothing. In particular, a lower value of the Laplace smoothing parameter implies a high confidence on the parameters learnt from the training set. The learnt parameters will surely have a good performance on the training set itself, but does not necessarily perform well on the test set. On the other hand, larger value of the smoothing parameter implies a conservative approach, which may have restricted performance on the training set, but will perform reasonably well on the test set, especially if the training set is noisy or sparse. Therefore, we observe the ideal behavior one would expect as an effect of smoothing. Further, the trend is observed for both MNB and MMNB across the entire range that is tested.

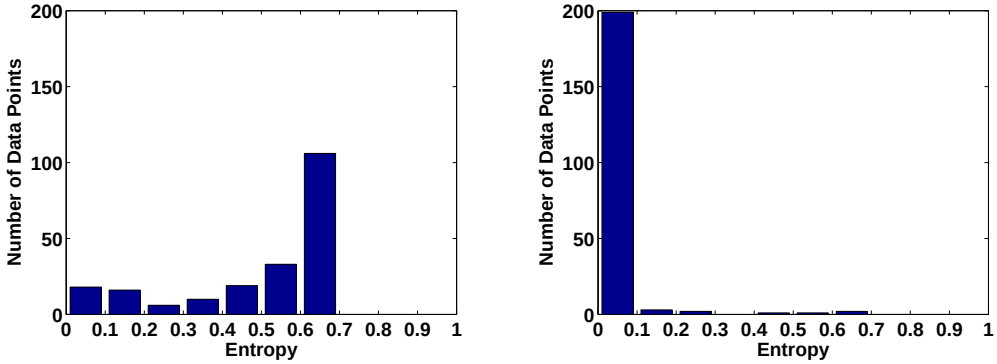
We ran extensive experiments for a range of values for the number of clusters k and the smoothing parameter ϵ. The overall results for the entire (k, ϵ) range on training and test sets are presented as perplexity surfaces in Figure 7. The key observations are as follows:

1. For the training set results in Figure 7(a), the perplexity surface for MMNB is almost always



(a) MMNB (b) NB

Figure 8: Histogram of mixed-membership entropy on Wdbc. For NB, almost all data points have a small mixed-membership entropy. For MMNB, most of the data points also have a small mixed-membership entropy, but there are a certain portion of exceptions.



(a) MMNB (b) NB

Figure 9: Histogram of mixed-membership entropy on Sonar. For NB, almost all data points have a small mixed-membership entropy. For MMNB, the mixed-membership entropy spreads over different ranges, mostly [0.6,0.7].

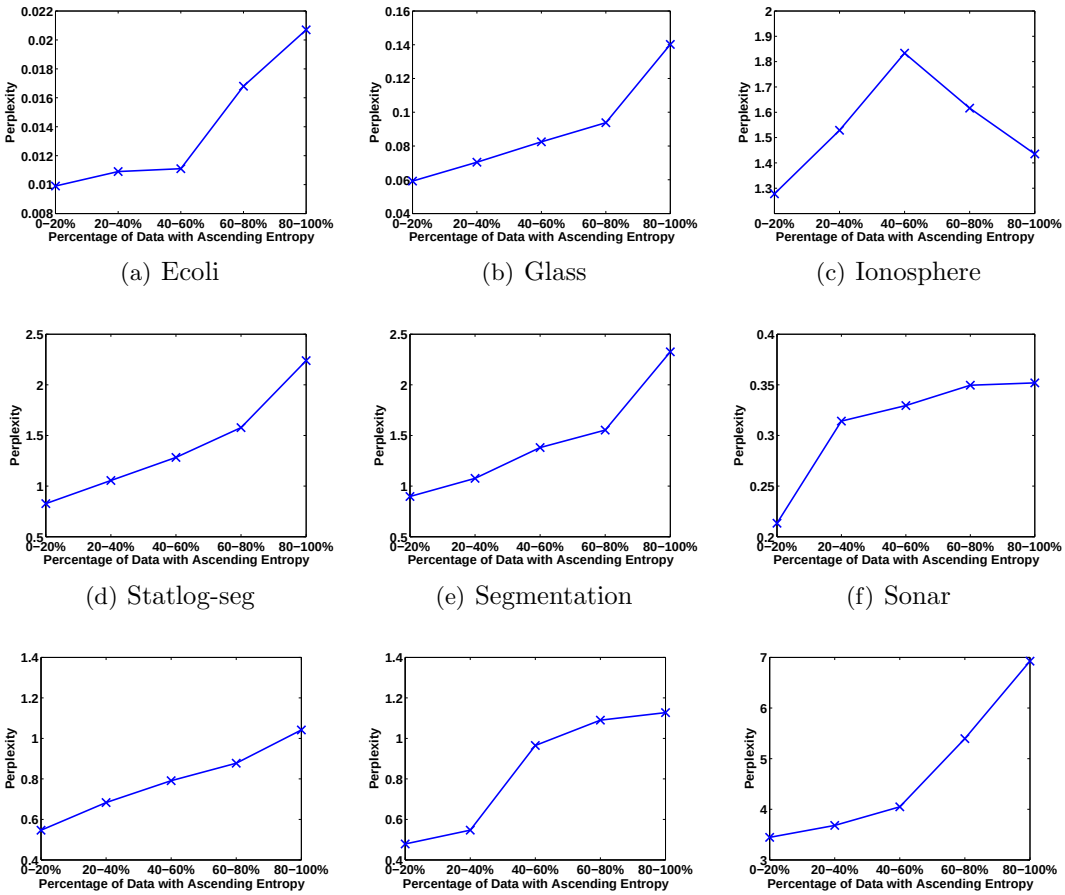
lower than that of MNB over the entire range. MNB tends to do marginally better than MMNB for a very large k and a very high ϵ.

2. Overall, the smoothing parameter has an adverse effect on the training set performance for both MMNB and MNB. Both models tend to perform better on the training set with a larger number of latent classes and a smaller value of the smoothing parameter.

3. For the test set results in Figure 7(b), MMNB achieves a lower perplexity than that of MNB for a smaller smoothing parameter. MNB performs marginally better than MMNB for high values of the smoothing parameter.

4. The test set performance of MMNB is very consistent across the entire range of (k, ϵ), which highlights the stability of the model.

5. MNB's test set performance for low ϵ values is poor, whereas the training set performance is very good, which is a clear indication of over-fitting.



(a) Ecoli (b) Glass (c) Ionosphere

(d) Statlog-seg (e) Segmentation (f) Sonar

(g) Vowel (h) Wdbc (i) Wine

Figure 10: Perplexities with ascending mixed-membership entropy on UCI dataset. Perplexities increase with ascending entropy on most of datasets.

Overall, MMNB demonstrates better performance on the training set and more consistent and mostly better performance on the test set. Its stability on test set across different choices of parameters demonstrates its modeling capabilities and makes it more suitable for real life tasks.

Mixed-Membership Assignments of MMNB

To obtain a better understanding of MMNB's behavior, we run more experiments on UCI data to study the mixed-membership of data points belonging to different clusters. In particular, we study the mixed membership using Shannon entropy of the distribution and compare this entropy between MMNB and NB. A low entropy implies almost hard clustering, whereas higher entropy implies a truely mixed-membership assignment of points. Figure 8 and 9 are two examples of mixed-membership entropy histograms on Wdbc and Sonar, where each bar shows the number of the data points falling into each range of entropies. MMNB and NB both have high accuracy on Wdbc, but low accuracy on Sonar. From the figures, we can see that NB's mixed-membership entropy on both Wdbc and Sonar mostly fall into the extreme low range of [0,0.1]. In comparison, for Wdbc where MMNB achieves a high micro-precision, MMNB also mostly generates low mixed-membership entropies, but there are a small portion of data points with relatively high entropies as [0.4,0.7]. Moreover, MMNB's entropies on Sonar, where its micro-precision is low, fall into

Num of Clusters	Training Set		Test set	
	MMNB	LDA	MMNB	LDA
5	1.7633±0.0034	98.1762±0.0386	1.7725±0.0231	98.2084±0.2624
10	1.7784±0.0028	98.3465±0.0385	1.8031±0.0219	98.3785±0.2577
15	1.8400±0.0079	98.5684±0.0303	1.8517±0.0224	98.5941±0.2597
20	1.8982±0.0018	98.6900±0.0304	1.8999±0.0202	98.7170±0.2600
25	1.9007±0.0022	98.8084±0.0305	1.9022±0.0200	98.8348±0.2599

Table 5: Perplexity Comparison for MMNB and LDA on Jester with varying number of clusters.

Num of Clusters	Training Set		Test set	
	MMNB	LDA	MMNB	LDA
5	1.7220±0.0080	466.1500±3.3531	1.9820±0.0495	515.4462±33.2887
10	1.6690±0.0044	430.8736±3.8621	2.0286±0.0477	502.5386±30.7209
15	1.6289±0.0046	407.4768±3.2146	2.0753±0.0521	506.3093±29.1375
20	1.5934±0.0027	397.6517±2.8421	2.1011±0.0506	513.4418±31.3045
25	1.5580±0.0030	388.1596±2.8644	2.1176±0.0550	525.1515±30.1515

Table 6: Perplexity comparison for MMNB and LDA on Movielens with varying number of clusters.

different ranges, mostly as high as [0.6,0.7]. Similar results are observed on other datasets, that is, the mixed-membership entropy from NB always falls into the range of [0,0,1], revealing that NB actually generates a somewhat "hard" clustering where each data point belonging to only one cluster with an extremely high probability. On the other hand, the mixed-membership entropy from MMNB always spreads over various ranges, meaning that MMNB generates a "soft" clustering.

NB's "hard" clustering suffers from at least two limitations: First, it puts a restriction on allowing one data point to belong to multiple clusters. Second, if the largest component of mixed membership does not correspond to the correct cluster, the hard clustering assigns the data point to a completely wrong cluster, since the posterior on even the second largest component is close to 0. In comparison, MMNB's soft clustering allows one data point to belong to multiple clusters with varying degrees. Therefore, even though the largest component does not match the right cluster, MMNB may still assign the data point to the right cluster with a certain probability, which means the log-loss [2] would be low. The conservative strategy always ensures MMNB to generate a more reasonable clustering result than NB.

To learn more properties of MMNB, we sort all the data points in the test sets in ascending order of their mixed-membership entropy, and divide the test sets evenly into five parts according to the ascending entropy, i.e., the first part contains the first 20% data points with the lowest entropy, the second part contains the second 20% data points with the second lowest entropy, and so on. We

Num of Clusters	Training Set		Test set	
	MMNB	LDA	MMNB	LDA
4	1.8668±0.0052	1503.0433±2.0196	2.0540±0.0142	1634.7652±3.7403
8	1.8114±0.0040	1362.1752±3.4512	2.1702±0.0193	2112.9106±21.5524
12	1.7537±0.0031	1385.4539±3.2186	2.2368±0.0179	1794.5040±13.1146
16	1.6892±0.0051	1375.2157±4.8707	2.2826±0.0222	1816.6249±8.7161
20	1.6368±0.0036	1354.8980±8.1904	2.3114±0.0200	1793.2581±17.642

Table 7: Perplexity Comparison for MMNB and LDA on Foodmart with varying number of clusters.

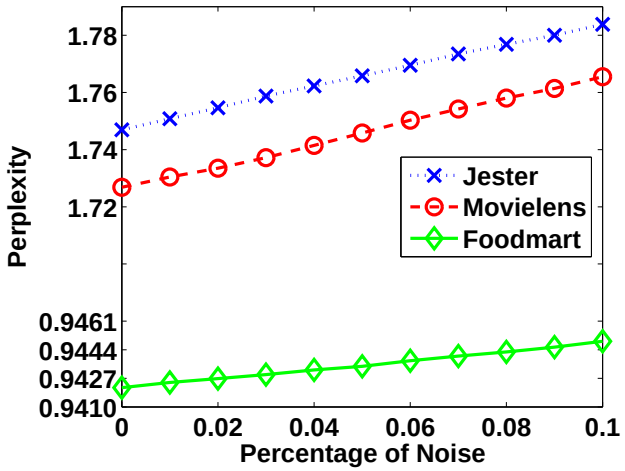


Figure 11: Perplexity curves for Movielens, Foodmart and Jester with increasing percentage of noise. Y-axis has been adjusted to accommodate all three curves. Perplexities on three datasets increase steadily with adding noise from 1% to 10%.

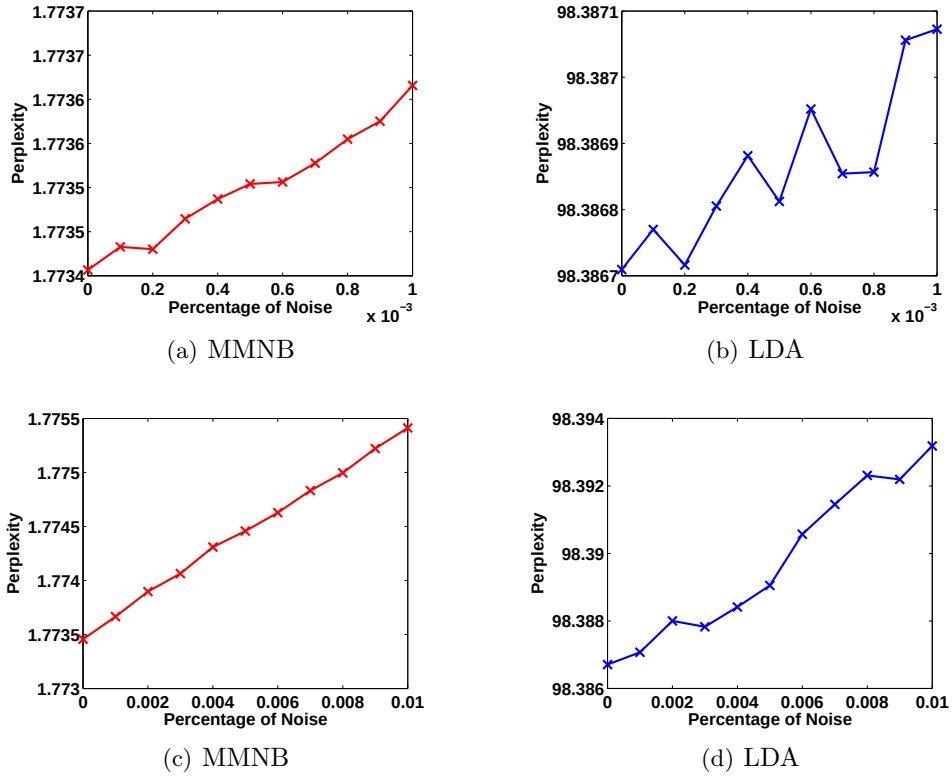
then calculate the perplexities on these five parts separately. *The hypothesis is* that the perplexity increases with ascending mixed-membership entropy, and ideally such increase is monotonic. In other words, if the model is confident about the cluster assignment of a point (low entropy), then the test-set perplexity on that point will be low. Figure 10 shows the curves as an average of 10-fold cross-validation on 9 UCI datasets. Surprisingly, the hypothesis is verified on almost all datasets, i.e., we observe increasing perplexity with higher mixed-membership entropy on all datasets except Ionosphere. Since the mixed-membership entropy measures the model's uncertainty of the result, and perplexity on test set measures how the model fits the test data, we learn MMNB's behavior: In general, the less confidence the model has with the clustering result (higher mixed-membership entropy), the worse performance it would get (higher perplexity).

6.4 MMNB vs. LDA

In this section, we compare MMNB with LDA on binarized Jester, Movielens and Foodmart. The lower perplexity achieved by MMNB verifies its flexibility compared with LDA.

6.4.1 Methodology

In principle, it is difficult to compare LDA with MMNB, because they are designed to deal with different data types, but we can apply these two models on binarized data to make the comparison possible. Given a binary data matrix, for MMNB, we choose the Bernoulli distribution as the generative model; for LDA, we consider the features as "words" in the dictionary, and consider the data points as the "documents" in the corpus. Then for each data point, all features with the feature value "1" become the "words" that appear in the "document". For example, in the binarized Movielens where all entries of 4 or 5 become 1, all movies that are rated 4 or 5 by the user i could be considered as the words appearing in document i. In that case, the "words" (movies) in the "document" (user) become the user's favorite movies. The same strategy was used by [7] to evaluate LDA on movie rating data.



(a) MMNB (b) LDA

(c) MMNB (d) LDA

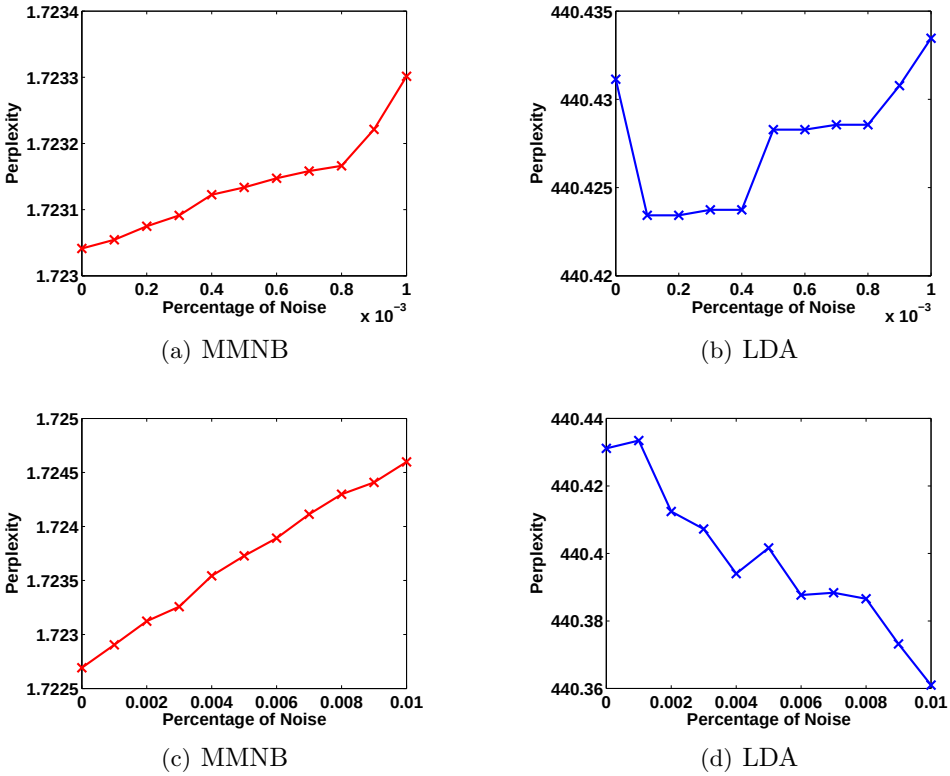
Figure 12: Perplexity curves of MMNB and LDA with increasing percentage of noise on binarized Jester. The perplexity of MMNB increases more steadily with increasing noise than that of LDA.

6.4.2 Results

The results of perplexities as an average of 10-fold cross-validation on Jester, Movielens, and Foodmart are presented in Table 5-Table 7 respectively. The perplexities of MMNB are orders of magnitudes lower than that of LDA across different number of clusters on all three datasets. MMNB's low perplexity on training set is possibly because it uses a separate distribution for each feature, instead of using a same distribution across all features as in LDA. Interestingly, despite large number of distributions, MMNB still has a much lower perplexity on test set. MMNB's lower perplexity seems to indicate that it fits the data and explains the data substantially better than LDA. However, one must be careful in drawing such conclusions since MMNB and LDA work on different variants of the data; we discuss this aspect further at the end of this subsection.

We also compare the performance of MMNB and LDA on test set as follows: We randomly hold out several entries in the data matrix as the test set X_{test} and train the model from the training set X_{train}. We test the perplexity on (X_{train}, X_{test}), as well as on $(X_{train}, \tilde{X}_{test})$, where $\tilde{X}_{test}$ is generated by simply flipping the entries 1 to 0 and 0 to 1 on randomly chosen $p\%$ of test data. We record the perplexities with the percentage of noise p increasing from 1% to 10% in steps of 1% and report the average perplexity of 10-fold cross-validation at each step. The perplexity curves for Movielens, Foodmart and Jester are shown in Figure 11.

At the starting point, with no noise, the perplexity is on the true test set X_{test}, and $\tilde{X}_{test}$ is further away from X_{test} with increasing percentage of noise added to it. *The hypothesis is* that if the model is good for the true data, as more test data are modified, the fitness between the



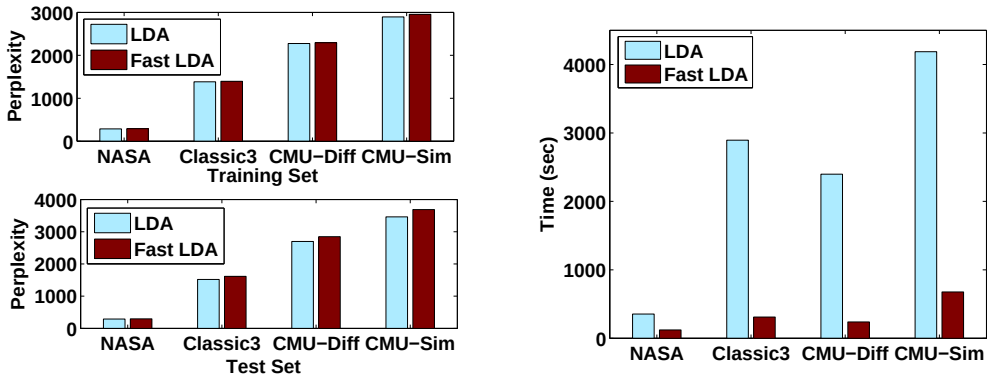
(a) MMNB (b) LDA

(c) MMNB (d) LDA

Figure 13: Perplexity curves of MMNB and LDA with increasing percentage of noise on binarized Movielens. The perplexity of MMNB increases consistently with increasing noise, but the perplexity of LDA goes down in (d).

model and data will decrease, and ideally such decrease is monotonic. As shown in Figure 11, all three perplexity lines go up steadily with an increasing percentage of test data modified. This is a surprisingly good result, implying that our model is able to detect increasing noise in the test set and convey the message through increasing perplexities. The most accurate result indicated by the model, i.e., the one with the lowest perplexity, is exactly the true test set at the starting point. Therefore, MMNB can potentially be used to accurately predict missing values in a matrix.

We add noise at a finer step of modifying 0.1% and 0.01% test data each time with ten steps respectively, and compare the prediction performance of MMNB with LDA. Figure 12 and 13 presents the results on Jester and Movielens. In both figures, the first row corresponds to adding noise at steps of 0.01% and the second row corresponds to adding noise at steps of 0.1%. The trends of the perplexity curves, instead of the absolute value of perplexities, demonstrate the prediction performance. On Jester, we can see that the perplexity curves for MMNB in both Figure 12(a) and 12(c) almost always go up with additional noise. However, the perplexity curves for LDA go up and down from time to time, especially in Figure 12(b), which means that sometimes LDA fits the data with more noise better than that with less noise, indicating a lower prediction accuracy compared with MMNB if used for prediction. The difference is even more distinct on the Movielens dataset. When adding noise at steps of 0.01%, MMNB's perplexity curve goes up steadily while LDA's perplexity curve drops dramatically at the beginning. When the step size increases to 0.1%, the perplexity curve of LDA even goes down as in Figure 13(d). These comparison show that



(a) Perplexity (b) Time

Figure 14: Comparison between LDA and Fast LDA in terms of perplexity and time.

MMNB's performance is more robust and consistent compared to LDA on the test set.

While extensive results give supportive evidence to MMNB's better performance, we should be cautious of the conclusion one can draw from the direct perplexity comparison between MMNB and LDA. Given a binary dataset, MMNB works on all non-missing entries, but LDA only works on the entries with value 1. Therefore, MMNB and LDA actually work on different data, and hence their perplexities cannot be compared directly. However, the comparison gives us a rough idea of these two algorithms' behavior, such as the distinct difference in perplexity ranges, similar perplexity trends with increasing number of clusters, etc.. Moreover, by comparing the perplexity trends with increasing noise instead of absolute perplexity values, it is shown that MMNB indeed has a better noise robustness performance on test set than LDA, no matter which part of data they perform on respectively.

6.5 Fast LDA vs. LDA

In this section, we demonstrate the advantage of fast variational inference used for Fast LDA versus the original one used in LDA [7]. We also present some comparisons between Fast MMNB with MMNB. The comparison is made in terms of running time and modeling performance. To evaluate the modeling performance, we again use perplexity. In addition, for text datasets, we also calculate the micro-precision and generate the word lists for topics. *The hypothesis is* that the Fast LDA would achieve a similar performance with LDA [7], but it would be much more computationally efficient.

Dataset	LDA	Fast LDA
NASA	91.4541%	92.8052%
Classic3	67.4757%	67.3330%
CMU-Diff	96.1481%	95.3063%
CMU-Sim	71.4000%	68.9000%

Table 8: Micro-precision comparison of LDA and Fast LDA on text datasets. The micro-precision obtained by LDA and Fast LDA are similar.

Four text datasets are used for comparing Fast LDA and LDA: NASA, Classic3, CMU-Diff and CMU-Sim. The comparisons of average perplexity and time over 10-fold cross-validation are

(a) LDA

Topic 1	Topic 2	Topic 3
runway	aircraft	passenger
approach	maintenance	flight
aircraft	engine	attendant
departure	zzz	captain
altitude	flight	seat
turn	minimum equipment list	told
time	check	asked
air traffic control	fuel	back
flight	time	attendants
tower	gear	aircraft

(b) Fast LDA

Topic 1	Topic 2	Topic 3
runway	aircraft	passenger
aircraft	maintenance	flight
approach	flight	attendant
flight	engine	capt
departure	minimum equipment list	told
time	zzz	seat
alt	check	asked
turn	time	aircraft
landing	control	back
air traffic control	crew	attendants

Table 9: Word list for three topics on NASA. The word lists from LDA and Fast LDA are qualitatively similar. Topic 1 is "flight crew", Topic 2 is "maintenance", and Topic 3 is "passenger".

presented in Figure 14. The time shown in the figure is the sum of two parts: training a model from the training set and applying it to the test set to calculate the perplexity. From the comparison, we observe very similar perplexities for Fast LDA and LDA on training sets, and a mildly higher perplexity for Fast LDA on test sets. The overall performance of these two models are quite close to each other. As for the running time, Figure 14(b) provides the supportive evidence that Fast LDA is 5-10 times faster than LDA. We also notice that Fast LDA on higher-dimensional dataset such as CMU-Sim ($d = 10083$) comparatively saves more time than on lower-dimensional dataset such as NASA ($d = 604$). As we have explained in Section 5, by introducing the fast variational inference, the number of variational parameters ϕ decreases from $\sum_{i=1}^{n} m_i$ to n, where m_i is the number of words for each document, and n is the total number of the documents. Accordingly, the optimization times over ϕ also decreases from $\sum_{i=1}^{n} m_i$ to n. Therefore, roughly, Fast LDA shows more advantage in terms of running time on higher-dimensional datasets.

We use 5% of the data as initialization, and run Fast LDA and LDA on the whole data sets to get micro-precision in Table 8 and word lists of topics for NASA and Classic3 as two examples in Table 9 and Table 10, where the words are listed with decreasing probabilities in each topic[7]. The micro-precision comparison shows that Fast LDA is always able to achieve a clustering accuracy which is close to LDA, and sometimes even slightly higher than LDA. Regarding the word lists for topics, we can make two observations based on Table 9 and Table 10. First, both LDA and Fast LDA generate appropriate word lists for the topics. We can map each list to the given topic without any effort. For example, in the result on NASA, Topic 1 is "flight crew", Topic 2 is "maintenance", and Topic 3 is "passenger". Second, the word lists from Fast LDA and LDA share most of the words. The main difference is just the rank of the words in the list.

We also compare Fast MMNB and MMNB on Jester, Movielens, and Foodmart. Figure 15(a) and 15(b) show the comparison of perplexity and running time respectively, both as an average of a 10-fold cross-validation. The perplexity obtained from Fast MMNB is quite close to that from MMNB, especially on training sets. Meanwhile, Figure 15(b) demonstrates that Fast MMNB is 5-10 times faster than MMNB.

[7] "zzz" in Table 9 is used to denote a name which should be kept secret.

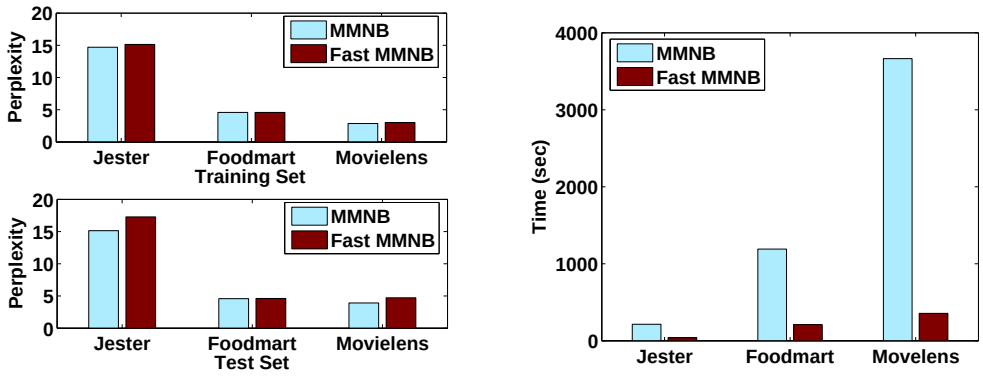
(a) LDA

Topic 1	Topic 2	Topic 3
information	patients	flow
library	cells	boundary
system	cases	pressure
data	normal	layer
libraries	growth	number
research	blood	mach
systems	found	results
retrieval	treatment	theory
science	children	heat
scientific	cell	method

(b) Fast LDA

Topic 1	Topic 2	Topic 3
information	patients	flow
library	cells	boundary
system	cases	pressure
libraries	normal	layer
data	growth	number
research	blood	mach
retrieval	treatment	results
systems	found	theory
science	children	shock
scientific	cell	heat

Table 10: Word list for three topics on Classic3. The word lists from LDA and Fast LDA are qualitatively similar. Topic 1 is "information retrieval", Topic 2 is "medicine", and Topic 3 is "aeronautics".



(a) Perplexity (b) Time

Figure 15: Comparison between MMNB and Fast MMNB in terms of perplexity and time.

7 Related Work

In this section, we present a brief discussion on the existing literatures related to mixed-membership models.

Probabilistic latent semantic indexing (pLSI) [21] is an extension of latent semantic index. pLSI represents each document as a mixing weights (discrete distribution) over a set of topics, i.e., each document has a mixed-membership belonging to different topics with certain degrees. pLSI also represents each topic as a distribution over all words in the dictionary. To generate each word in the document, pLSI first picks a topic based on the mixed-membership of the document, then generates the word from the distribution of that topic over the words.

While pLSI defines a proper generative model for observed data, it does not have a generative model for unseen data. In other words, there is only a finite set (the set of the documents in the training set) of the mixed-memberships over the topics, but no generative model for these mixed-memberships. Latent Dirichlet allocation (LDA) [7] relaxes this restriction by introducing a Dirichlet prior on the topic simplex such that the mixed-membership over topics could be generated from this prior. As an application of LDA, [19] uses a full Bayesian model to analyze abstracts from

Proceedings of the National Academy of Sciences (PNAS). It not only gives the mixed-membership of the abstracts belonging to multiple topics, but also gives the correlation between topics and the evolution of popular topics over years by analyzing mixed-memberships.

[14] propose a more general mixed-membership model which contains four levels, as four levels in a generative Bayesian model. The model generalizes LDA in the sense that in principle, it allows the mixed-membership to be generated from various distributions other than Dirichlet distribution, and each feature to be generated from various distributions other than discrete distribution. However, it is different from MMNB in that it still assumes that all features share a same distribution. The authors apply the model to topic modeling for scientific publications on PNAS. The model takes both the words and references into consideration by choosing an appropriate distribution for the references.

Recently, considerable amount of work has been done on mixed-membership of relational data. [1] proposes mixed-membership stochastic blockmodels to deal with binary relationships between the objects within a group. [38] proposes Bayesian co-clustering which generates mixed-membership by taking the relationship of various types between the objects in two groups respectively. The application of the mixed-membership for relational data includes protein-protein interaction analysis [1], social network analysis [23], etc..

One of the most recent progresses on mixed-membership models is Bayesian partial membership model (BPM) [20]. BPM is a full Bayesian model which introduces a Dirichlet distribution on the Dirichlet prior over mixed-membership as well as a conjugate prior on component distribution, and the component distribution could be any exponential family distribution. The main difference between BPM and MMNB is that unlike MMNB, BPM does not assume a factorization over the features of a data point.

8 Conclusion

In this paper, we propose a family of mixed-membership naive Bayes (MMNB) models. Such models extend the popular naive Bayes (NB) models to work with sparse observations, by marginalizing over all missing features. In addition, they take advantage of the machinery of hierarchical Baysian modeling to allow NB models to generate mixed-memberships for the data points. [7] had suggested that such an extension will be possible due to the modularity of latent Dirichlet allocation (LDA). In this paper, we demonstrate how powerful such an extension can be in the context of NB models, while advancing the state-of-the-art on NB as well as LDA. Moreover, the new fast variational inference algorithms proposed ensure the scalability of MMNB models. When applied in the context of topic modeling, the same ideas lead to a substantially more efficient algorithm for LDA. Extensive experiments on a variety of datasets demonstrate that MMNB has a better performance than NB in terms of clustering accuracy, predictive perplexity, as well as stability. Although MMNB and LDA are designed for different types of data, the comparisons on binary datasets show that MMNB is more robust. Further, Fast LDA exhibits a substantial improvement in computational efficiency as compared to LDA on all text datasets considered.

When applying the model to real applications, for example, movie recommendation systems, one important problem that still needs to be solved is prediction, i.e., predicting user's ratings on certain movies. A brute force way would be to try all possible ratings and pick the one with the lowest perplexity. However, the cost of such computation would be exponential in the number of ratings to be predicted, since the ratings are not independent according to the model. Such a

problem motivates further study on how to do prediction efficiently using such mixed-membership Bayesian models. Besides, it will be important to investigate automatic model selection approaches for MMNB models, such as choosing the number of latent clusters, and choosing appropriate exponential family for each feature.

Acknowledgements: The research was supported by NASA grant NNX08AC36A and NSF grant IIS-0812183.

A Variational Inference and Parameter Estimation

In this appendix, we give derivations for variational inference algorithms in Section 4 and 5. In Appendix A.1, we give the derivation for MMNB as a direct generalization of the inference in LDA, and in Appendix A.2, we give the derivation for Fast MMNB and Fast LDA.

A.1 MMNB

Given a data point $\mathbf{x}$, since a direct computation of $\log p(\mathbf{x}|\alpha, \Theta)$ is intractable, following [7], we introduce for each data point a variational distribution (Figure 3(a))

$$q(\pi, \mathbf{z}|\gamma, \phi) = q(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^{d} q(z_j|\phi_j) \tag{40}$$

as a surrogate for the posterior distribution $p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x})$, where γ is a Dirichlet parameter over π and $\phi = \{\phi_j, [j]_1^d, \exists x_j\}$ are discrete parameters over the component z for each of non-missing features. By applying Jensen's inequality, we have [7]:

$$\begin{aligned}
\log p(\mathbf{x}|\alpha, \Theta) =& \log \int_\pi \sum_{\mathbf{z}} p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta) d\pi \\
=& \log \int_\pi \sum_{\mathbf{z}} q(\pi, \mathbf{z}|\gamma, \phi) \frac{p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)}{q(\pi, \mathbf{z}|\gamma, \phi)} d\pi \\
\geq& \int_\pi \sum_{\mathbf{z}} q(\pi, \mathbf{z}|\gamma, \phi) \log \frac{p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)}{q(\pi, \mathbf{z}|\gamma, \phi)} d\pi \\
=& \int_\pi \sum_{\mathbf{z}} q(\pi, \mathbf{z}|\gamma, \phi) \log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta) d\pi - \int_\pi \sum_{\mathbf{z}} q(\pi, \mathbf{z}|\gamma, \phi) \log q(\pi, \mathbf{z}|\gamma, \phi) d\pi \\
=& E_q[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)] + H(q(\pi, \mathbf{z}|\gamma, \phi)) \ .
\end{aligned} \tag{41}$$

Therefore (41) gives a lower bound to $\log p(\mathbf{x}|\alpha, \Theta)$. For each data point $\mathbf{x}_i$, denoting the lower bound with $L(\gamma_i, \phi_i; \alpha, \Theta)$, we can expand it as

$$\begin{aligned}
L(\gamma_i, \phi_i; \alpha, \Theta) =& E_q[\log p(\pi_i|\alpha)] + E_q[\log p(\mathbf{z}_i|\pi_i)] + E_q[\log p(\mathbf{x}_i|\Theta, \mathbf{z}_i)] \\
& - E_q[\log q(\pi_i|\gamma_i)] - E_q[\log q(\mathbf{z}_i|\phi_i)] \ .
\end{aligned} \tag{42}$$

Each term in $L(\gamma_i, \phi_i; \alpha, \Theta)$ could be further expanded as follows:

$$\begin{aligned}
E_q[\log p(\pi_i|\alpha)] &= \log\Gamma(\sum_{c=1}^{k}\alpha_c) - \sum_{c=1}^{k}\log\Gamma(\alpha_c) + \sum_{c=1}^{k}(\alpha_c - 1)(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il})) \\
E_q[\log p(\mathbf{z}_i|\pi_i)] &= \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d}\sum_{c=1}^{k}\phi_{ijc}(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il})) \\
E_q[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] &= \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d}\sum_{c=1}^{k}\phi_{ijc}\log p_{\psi_j}(x_{ij}|\theta_{jc}) \\
E_q[\log q(\pi_i|\gamma_i)] &= \log\Gamma(\sum_{c=1}^{k}\gamma_{ic}) - \sum_{c=1}^{k}\log\Gamma(\gamma_{ic}) + \sum_{c=1}^{k}(\gamma_{ic} - 1)(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il})) \\
E_q[\log q(\mathbf{z}_i|\phi_i)] &= \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d}\sum_{c=1}^{k}\phi_{ijc}\log\phi_{ijc}\ ,
\end{aligned}$$

where γ_{ic} is the c^{th} component of the variational Dirichlet distribution for the i^{th} data point, ϕ_{ijc} is the c^{th} component of the variational discrete distribution of the j^{th} feature in the i^{th} data point, and Ψ is the digamma function, i.e., the first derivative of the log Gamma function.

A.1.1 Variational Inference

To obtain the variational parameters, we first maximize $L(\gamma_i, \phi_i; \alpha, \beta)$ with respect to ϕ_{ijc}. Since it is a constrained maximization under the constraint $\sum_{c=1}^{k}\phi_{ijc} = 1$, we construct the Lagrangian by isolating the terms containing ϕ_{ijc} and adding the Lagrange multipliers

$$L_{[\phi_{ijc}]} = \phi_{ijc}\left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il}) - \log\phi_{ijc} + \log p_{\psi_j}(x_{ij}|\theta_{jc})\right) + \lambda_{ij}(\sum_{c=1}^{k}\phi_{ijc} - 1)\ ,$$

where λ_{ij} is the Lagrangian multiplier. Taking derivative with respect to ϕ_{ijc} and setting it to zero, we have

$$\phi_{ijc} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il})\right) p_{\psi_j}(x_{ij}|\theta_{jc})\ ,\ [i]_1^k,\ [j]_1^d\ ,\ [c]_1^k\ .$$

Second, we maximize $L(\gamma_i, \phi_i; \alpha, \Theta)$ with respect to γ_{ic}. The terms containing γ_{ic} are

$$L_{[\gamma_{ic}]} = (\alpha_c + \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d}\phi_{ijc} - \gamma_{ic})(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k}\gamma_{il})) - \log\Gamma(\sum_{c=1}^{k}\gamma_{ic}) + \log\Gamma(\gamma_{ic})\ .$$

Taking derivative with respect to γ_{ic} and setting it to zero, we get

$$\gamma_{ic} = \alpha_c + \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d}\phi_{ijc}\ ,\ [i]_1^k,\ [c]_1^k\ .$$

A.1.2 Parameter Estimation

For variational inference, we consider each single data point separately to get variational parameters for each of them. In this section, we consider all data points together to obtain the estimate for the model parameters. The overall log-likelihood of the whole dataset $\mathcal{X} = \{\mathbf{x}_i, [i]_1^n\}$ is the summation of log-likelihoods for all individual data points, accordingly, the lower bound of log-likelihood of the whole dataset is the summation of the lower bounds (42) for all data points, i.e.,$\sum_{i=1}^n L(\gamma_i, \phi_i; \alpha, \Theta)$.

To maximize the lower bound of log-likelihood with respect to θ_{jc}, the terms containing θ_{jc} are given by:

$$L_{[\theta_{jc}]} = \sum_{\substack{i=1 \\ \exists x_{ij}}}^{n} \phi_{ijc} \log p_{\psi_j}(x_{ij}|\theta_{jc}) \ .$$

Following [5], any regular exponential family distribution in the form of

$$p_\psi(x|\theta) = \exp(\langle x, \theta \rangle - \psi(\theta)) p_0(x)$$

can be expressed in terms of its expectation parameter τ as

$$p(x|\tau) = \exp(-d_f(x, \tau)) b_f(x) \ ,$$

where $b_f = \exp(f(x)) p_0(x)$, $d_f(\cdot, \cdot)$ is the Bregman divergence determined by the function f, which is the conjugate of the cumulant function ψ of the family, and $\tau = E[X] = \nabla\psi(\theta)$ with θ the natural parameter. From this perspective, let s_{ij} denote the sufficient statistics for x_{ij}, then the estimation for the mean τ_{jc} of the j^{th} feature and the c^{th} component is given by the weighted average of s_{ij} as

$$\tau_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} s_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}} \ , \ [j]_1^d \ , \ [c]_1^k \ ,$$

and by conjugacy, we have

$$\theta_{jc} = \nabla f_j(\tau_{jc}) \ .$$

In particular, for Gaussian distribution, we have

$$L_{[\mu_{jc}, \sigma_{jc}^2]} = \sum_{\substack{i=1 \\ \exists x_{ij}}}^{n} \phi_{ijc} \left(-\frac{(x_{ij} - \mu_{jc})^2}{2\sigma_{jc}^2} - \log \sqrt{2\pi\sigma_{jc}^2} \right) \ .$$

Taking derivative with respect to μ_{jc} and σ_{jc}^2, and setting them to zero, we have

$$\begin{aligned} \mu_{jc} &= \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} x_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}} \\ \sigma_{jc}^2 &= \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} (x_{ij} - \mu_{jc})^2}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}} \ , \ [j]_1^d \ , \ [c]_1^k \ . \end{aligned}$$

For discrete distribution, we construct the Lagrangian as

$$L_{[p_{jc}(r)]} = \sum_{i=1}^{n} \phi_{ijc} \sum_{r=1}^{r_j} \mathbb{1}(x_{ij} = r) \log p_{jc}(r) + \lambda_{jc} (\sum_{r=1}^{r_j} p_{jc}(r) - 1) \ ,$$

where λ_{jc} is the Lagrange multiplier. Taking derivative with respect to $p_{jc}(r)$ and setting it to zero, we have

$$p_{jc}(r) \propto \sum_{i=1}^{n} \mathbb{1}(x_{ij} = r)\phi_{ijc} \ , \ [j]_1^d \ , \ [c]_1^k \ , [r]_1^{r_j} \ .$$

To maximize the lower bound with respect to α, the terms containing α are given by:

$$L_{[\alpha]} = \sum_{i=1}^{n} \left(\log \Gamma(\sum_{l=1}^{k} \alpha_l) - \sum_{c=1}^{k} \log \Gamma(\alpha_c) + \sum_{c=1}^{k} (\alpha_c - 1)(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il})) \right) \ .$$

Taking derivative with respect to α yields the gradient $g(\cdot)$ as

$$\frac{\partial L}{\partial \alpha_c} = \sum_{i=1}^{n} \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il}) \right) + n \left(\Psi(\sum_{c=1}^{k} \alpha_c) - \Psi(\alpha_c) \right) \ . \tag{43}$$

The derivation depends on $\{\alpha_l, [l]_1^k, l \neq c\}$, so there is no closed form solution for α_c. Following [7], we use Newton-Raphson algorithm to update α_c iteratively, where,

$$\frac{\partial L}{\partial \alpha_c \alpha_c} = n\Psi'(\sum_{c=1}^{k} \alpha_c) - n\Psi'(\alpha_c) \tag{44}$$

$$\frac{\partial L}{\partial \alpha_c \alpha_l} = n\Psi'(\sum_{c=1}^{k} \alpha_c) \ (l \neq c) \ , \tag{45}$$

so the Hessian matrix $H(\cdot)$ has (44) on diagonal and (45) off diagonal.

Given $g(\cdot)$ and $H(\cdot)$, Newton-Raphson algorithm finds the optimal solution by using the following updating equation:

$$\alpha' = \alpha + H(\alpha)^{-1} g(\alpha) \ .$$

In particular, given $g(\cdot)$ and $H(\cdot)$ as in (43) and (44, 45) respectively, the update equation for α_c is given by

$$\alpha_c' = \alpha_c - \frac{g_c - u}{h_c} \ , \ [c]_1^k \ , \tag{46}$$

where

$$\begin{aligned} g_c &= n \left(\Psi \left(\sum_{l=1}^{k} \alpha_l \right) - \Psi(\alpha_c) \right) + \sum_{i=1}^{n} \left(\Psi(\gamma_{ic}) - \Psi \left(\sum_{c=1}^{k} \gamma_{ic} \right) \right) \\ h_c &= -n\Psi'(\alpha_c) \\ u &= \frac{\sum_{c=1}^{k} g_c / h_c}{w^{-1} + \sum_{c=1}^{k} h_c^{-1}} \\ w &= n\Psi'(\sum_{c=1}^{k} \alpha_c) \ . \end{aligned}$$

The problem with the update equation (46) is that it ignores the fact that α has a constraint of $\alpha_c > 0$. Iterating using (46) sometimes takes the updated value outside the feasible range.

Therefore, we are using an adaptive line search in the updating direction. The update equation is given by

$$\alpha_c' = \alpha_c - \eta \frac{g_c - u}{h_c} \ , \ [c]_1^k \ . \tag{47}$$

Multiplying the second term by η, we are performing a line search to prevent α_c to go out of the feasible range ($\alpha_c > 0$). At each updating step, we first let η equal to 1, in that case, (47) becomes (46). After each iteration, if α_c is inside the feasible range, we go on to the next iteration, otherwise, we decrease η by a factor of 0.5 until α_c becomes valid. The objective function is guaranteed to be improved since we are not changing the update direction but only the scale.

A.2 Fast MMNB

In this section, we give the derivation for Fast MMNB by introducing a new variational distribution for each data point, given by (Figure 3(b))

$$q'(\pi, \mathbf{z}|\gamma, \phi) = q'(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^{d} q'(z_j|\phi) \ , \tag{48}$$

where γ is the parameter for variational Dirichlet distribution over π, and ϕ is the parameter for variational discrete distribution over all latent components z for all features. Again, by applying Jensen's inequality, we obtain the lower bound for $\log p(\mathbf{x}|\alpha, \Theta)$ as

$$\log p(\mathbf{x}|\alpha, \Theta) \geq E_{q'}[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)] - E_{q'}[\log q'(\pi, \mathbf{z}|\gamma, \phi)] \ .$$

Denoting the lower bound for $\mathbf{x}_i$ with $L(\gamma_i, \phi_i; \alpha, \Theta)$, it could be expanded as

$$\begin{aligned} L(\gamma_i, \phi_i; \alpha, \Theta) =& E_{q'}[\log p(\pi_i|\alpha)] + E_{q'}[\log p(\mathbf{z}_i|\pi_i)] + E_{q'}[\log p(\mathbf{x}_i|\Theta, \mathbf{z}_i)] \\ & - E_{q'}[\log q'(\pi_i|\gamma_i)] - E_{q'}[\log q'(\mathbf{z}_i|\phi_i)] \ , \end{aligned} \tag{49}$$

where

$$E_{q'}[\log p(\pi_i|\alpha)] = \log \Gamma(\sum_{c=1}^{k} \alpha_c) - \sum_{c=1}^{k} \log \Gamma(\alpha_c) + \sum_{c=1}^{k} (\alpha_c - 1)(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il})) \tag{50}$$

$$E_{q'}[\log p(\mathbf{z}_i|\pi_i)] = m_i \sum_{c=1}^{k} \phi_{ic} (\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il})) \tag{51}$$

$$E_{q'}[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] = \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d} \sum_{c=1}^{k} \phi_{ic} \log p_{\psi_j}(x_{ij}|\theta_{jc}) \tag{52}$$

$$E_{q'}[\log q'(\pi_i|\gamma_i)] = \log \Gamma(\sum_{c=1}^{k} \gamma_{ic}) - \sum_{c=1}^{k} \log \Gamma(\gamma_{ic}) + \sum_{c=1}^{k} (\gamma_{ic} - 1)(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il})) \tag{53}$$

$$E_{q'}[\log q'(\mathbf{z}_i|\phi_i)] = m_i \sum_{c=1}^{k} \phi_{ic} \log \phi_{ic} \ , \tag{54}$$

where γ_{ic} and ϕ_{ic} are the variational Dirichlet distribution and discrete distribution for the c^{th} component of $\mathbf{x}_i$ respectively, and m_i is the number of non-missing entries in each data point $\mathbf{x}_i$.

A.2.1 Variational Inference

First, We maximize $L(\gamma_i, \phi_i; \alpha, \beta)$ with respect to ϕ_{ic}. Similar with Appendix A.1, it is a constrained maximization under the constraint $\sum_{c=1}^{k} \phi_{ic} = 1$, we construct the Lagrangian as:

$$L_{[\phi_{ic}]} = m_i \phi_{ic} \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il}) - \log \phi_{ic} \right) + \sum_{\substack{j=1 \\ \exists x_{ij}}}^{d} \phi_{ic} \log p_{\psi_j}(x_{ij}|\theta_{jc}) + \lambda_i (\sum_{c=1}^{k} \phi_{ic} - 1) \ ,$$

where λ_i is the Lagrange multiplier. Taking derivative with respect to ϕ_{ic} and setting it to zero, we have

$$\phi_{ic} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^{k} \gamma_{il} \right) \right) \left(\prod_{\substack{j=1 \\ \exists x_{ij}}}^{d} p_{\psi_j}(x_{ij}|\theta_{jc}) \right)^{1/m_i} \quad , \ [i]_1^k \ , \ [c]_1^k \ .$$

Second, we maximize $L(\gamma_i, \phi_i; \alpha, \Theta)$ with respect to γ_{ic}. The terms containing γ_{ic} are:

$$L_{[\gamma_{ic}]} = (\alpha_c + m_i \phi_{ic} - \gamma_{ic})(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il})) - \log \Gamma(\sum_{c=1}^{k} \gamma_{ic}) + \log \Gamma(\gamma_{ic}) \ .$$

Taking derivative with respect to γ_{ic} and setting it to zero, we get

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \ , \ [i]_1^k, \ [c]_1^k \ .$$

A.2.2 Parameter Estimation

Similar as Appendix A.1.2, we consider the whole dataset $\mathcal{X} = \{\mathbf{x}_i, [i]_1^n\}$ together for parameter estimation. The lower bound of the log-likelihood on $\mathcal{X}$ is $\sum_{i=1}^{n} L(\gamma_i, \phi_i; \alpha, \Theta)$ To maximize with respect to θ_{jc}, the terms containing θ_{jc} are

$$L_{[\theta_{jc}]} = \sum_{\substack{i=1 \\ \exists x_{ij}}}^{n} \phi_{ic} \log p_{\psi_j}(x_{ij}|\theta_{jc}) \ .$$

Again, from Bregmam divergence perspective, the estimation of expectation τ_{jc} is given by

$$\tau_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^{n} \phi_{ic} s_{ij}}{\sum_{i=1, \exists x_{ij}}^{n} \phi_{ic}} \ , \ [j]_1^d \ , \ [c]_1^k \ ,$$

where s_{ij} is the sufficient statistics.

In particular, for Gaussian, we have

$$L_{[\mu_{jc}, \sigma_{jc}^2]} = \sum_{\substack{i=1 \\ \exists x_{ij}}}^{n} \phi_{ic} \left(-\frac{(x_{ij} - \mu_{jc})^2}{2\sigma_{jc}^2} - \log \sqrt{2\pi\sigma_{jc}^2} \right) \ .$$

By taking derivative with respect to μ_{jc} and σ^2_{jc} and setting them to zero, we get

$$\begin{aligned}\mu_{jc} &= \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic} x_{ij}}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}} \\ \sigma^2_{jc} &= \frac{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}(x_{ij}-\mu_{jc})^2}{\sum_{i=1,\exists x_{ij}}^{n} \phi_{ic}} \ , \ [j]_1^d \ , \ [c]_1^k \ .\end{aligned}$$

For discrete case, we construct the Lagrangian as

$$L_{[p_{jc}(r)]} = \sum_{i=1}^{n} \phi_{ic} \sum_{r=1}^{r_j} \mathbb{1}(x_{ij}=r) \log p_{jc}(r) + \lambda_{jc}(\sum_{r=1}^{r_j} p_{jc}(r) - 1) \ ,$$

where λ_{jc} is the Lagrange multiplier. Taking derivative with respect to $p_{jc}(r)$ and setting it to zero, we have

$$p_{jc}(r) \propto \sum_{i=1}^{n} \mathbb{1}(x_{ij}=r)\phi_{ic} \ , \ [j]_1^d \ , \ [c]_1^k \ , \ [r]_1^{r_j} \ .$$

The update equation for α is the same with (47).

A.3 Fast LDA

The variational distribution introduced for Fast LDA is the same as (48). Similarly, by applying Jensen's inequality, the lower bound $L(\phi_i, \gamma_i; \alpha, \beta)$ of the log-likelihood for each document $\mathbf{w}_i$ is given by

$$\begin{aligned}L(\gamma_i, \phi_i; \alpha, \Theta) =& E_{q'}[\log p(\pi_i|\alpha)] + E_{q'}[\log p(\mathbf{z}_i|\pi_i)] + E_{q'}[\log p(\mathbf{w}_i|\beta, \mathbf{z}_i)] \\ & - E_{q'}[\log q'(\pi_i|\gamma_i)] - E_{q'}[\log q'(\mathbf{z}_i|\phi_i)] \ , \end{aligned} \tag{55}$$

where the terms 1, 2, 4, 5 are the same with (50), (51), (53) and (54) respectively, and the term 3 could be expanded as:

$$E_{q'}[\log p(\mathbf{w}_i|\beta, \mathbf{z}_i)] = \sum_{j=1}^{m_i} \sum_{c=1}^{k} \sum_{v=1}^{V} \phi_{ic} w_{ij}^v \log \beta_{cv} \ .$$

A.3.1 Variational Inference

To maximize with respect to ϕ_{ic}, noticing $\sum_{v=1}^{V} \beta_{cv} = 1$, we construct the Lagrangian as

$$L_{[\phi_{ic}]} = m_i \phi_{ic} \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il}) - \log \phi_{ic} \right) + \sum_{j=1}^{m_i} \sum_{v=1}^{V} \phi_{ic} w_{ij}^v \log \beta_{cv} + \lambda_i (\sum_{c=1}^{k} \phi_{ic} - 1) \ ,$$

where λ_i is the Lagrange multiplier. Taking derivative with respect to ϕ_{ic} and setting it to zero, the update equation for ϕ_{ic} is given by

$$\phi_{ic} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^{k} \gamma_{il}) + \frac{1}{m_i} \sum_{j=1}^{m_i} \sum_{v=1}^{V} w_{ij}^v \log \beta_{cv} \right) \ , \ [i]_1^n \ , \ [c]_1^k \ ,$$

The solution for γ_{ic} is the same with Fast MMNB, that is,

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \ , \ [i]_1^k, \ [c]_1^k \ .$$

A.3.2 Parameter Estimation

To maximize with respect to β_{cv}, we construct the Lagrangian as

$$L_{[\beta_{cv}]} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \phi_{ic} w_{ij}^v \log \beta_{cv} + \lambda_c (\sum_{v=1}^{V} \beta_{cv} - 1) \ .$$

Taking derivative with respect to β_{cv} yields

$$\beta_{cv} \propto \sum_{i=1}^{n} \phi_{ic} \sum_{j=1}^{m_i} w_{ij}^v \ , \ [c]_1^k \ , \ [v]_1^V \ .$$

The update equation for α is the same with (47).

References

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1823–1856, 2008.

[2] Y. Altun, M. Johnson, and T. Hofmann. Loss functions and optimization methods for discriminative learning of label sequences. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2003.

[3] A. Banerjee, I. Dhillon, J. Ghosh, and S. Merugu. An information theoretic analysis of maximum likelihood mixture estimation for exponential families. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[4] A. Banerjee, C. Krumpelman, S. Basu, R. Mooney, and J. Ghosh. Model based overlapping clustering. In *Proceedings of the 11th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 532–537, 2005.

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

[6] O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. John Wiley and Sons, 1978.

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

[8] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[9] B. de Finetti. *Theory of probability*. John Wiley & Sons Ltd., 1990.

[10] M. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, 1970.

[11] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 89–98, 2003.

[12] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning Journal*, 29:103–130, 1997.

[13] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[14] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. In *Proceedings of the National Academy of Science (PNAS)*, pages 5220–5227, 2004.

[15] Q. Fu and A. Banerjee. Multiplicative mixture models for overlapping clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 791–796, 2008.

[16] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2003.

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

[18] Z. Ghahramani. Factorial learning and the EM algorithm. In *Proceedings of the 8th Annual Conference on Neural Information Processing Systems (NIPS)*, 1995.

[19] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Science (PNAS)*, 101(1):5228–5225, 2004.

[20] K. Heller, S. Williamson, and Z. Ghahramani. Statistical models for partial membership. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 392–399, 2008.

[21] T. Hoffman. Probabilistic latent semantic indexing. In *Proceedings of the 15th Conference in Uncertainty in Artificial Intelligence (UAI)*, 1999.

[22] T. Jaakkola. *Algorithms for Clustering Data*. MIT Press, 2000.

[23] P. Koutsourelakis and T. Eliassi-Rad. Finding mixed-memberships in social networks. In *Proceedings of National Conference on Artificial Intelligence, Spring Symposium on Social Information Processing*, 2008.

[24] K. Lang. News weeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, 1995.

[25] G. McLachlan and T. Krishnan. *The EM algorithm and Extensions*. Wiley-Interscience, 1996.

[26] T. Minka. Estimating a Dirichlet distribution. Technical report, Massachusetts Institute of Technology, 2003.

[27] T. Mitchell, R. Hutchinson, R. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

[28] D. Modha and S. Spangler. Feature weighting in k-means clustering. *Machine Learning*, 52(3):217–237, 2003.

[29] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

[30] A. Ng and M. Jordan. On discrminative vs generative classifiers: A comparison of logistic regression and naive Bayes. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems (NIPS)*, 2001.

[31] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[32] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 437–444, 2001.

[33] R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[34] J. Reichardt and S. Bornholdt. Clustering of sparse data via network communitiesa prototype study of a large online market. *Journal of Statistical Mechanics: Theory and Experiment*, 2007.

[35] E. Saund. Unsupervised learning of mixtures of multiple causes in binary data. In *Proceedings of the 7th Annual Conference on Neural Information Processing Systems (NIPS)*, 1994.

[36] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In *Proceedings of 8th Pacific Symposium on Biocomputing (PSB)*, 2003.

[37] M. Shahami, M. Hearst, and E. Saund. Applying the multiple cause model to text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 435–443, 1997.

[38] H. Shan and A. Banerjee. Bayesian co-clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 530–539, 2008.

[39] A. Strehl and J. Ghosh. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of the 7th International Conference on High Performance Computing*, 2000.

[40] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. Technical Report TR 649, Dept. of Statistics, University of California at Berkeley, 2003.

[41] M. Yousef, S. Jung, A. Kossenkov, L. Showe, and M. Showe. Naive Bayes for microRNA target predictions machine learning for microRNA targets. *Bioinformatics*, 23(22):2987–2792, 2007.