

Mixed-membership naive Bayes models

Hanhuai Shan · Arindam Banerjee

Received: 24 December 2009 / Accepted: 3 August 2010
© The Author(s) 2010

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 is the family of naive Bayes (NB) 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 do not allow data points to belong to different component clusters with varying degrees, i.e., mixed memberships, which puts a restriction on their modeling ability. In this paper, we propose mixed-membership naive Bayes (MMNB) models. On one hand, MMNB can be viewed as a generalization of NB by putting a Dirichlet prior on top to allow mixed memberships. On the other hand, MMNB can also be viewed as a generalization of latent Dirichlet allocation (LDA) with the ability to handle heterogeneous feature vectors with different types of features, e.g., real, categorical, etc.. We propose two variational inference algorithms to learn MMNB models. The first one is based on ideas originally used in LDA, and the second one uses substantially fewer variational parameters, leading to a significantly faster algorithm. Further, we extend MMNB/LDA to discriminative mixed-membership models for classification by suitably combining MMNB/LDA with multi-class logistic regression. The efficacy of the proposed mixed-membership models is demonstrated by extensive experiments on several datasets, including UCI benchmarks, recommendation systems, and text datasets.

Responsible editor: Charles Elkan.

H. Shan (✉) · A. Banerjee
Department of Computer Science & Engineering, University of Minnesota,
Twin Cities, Minneapolis, MN, USA
e-mail: shan@cs.umn.edu

A. Banerjee
e-mail: banerjee@cs.umn.edu

Keywords Naive Bayes · Latent Dirichlet allocation · Mixed-membership · Generative models · Variational inference · Logistic regression

1 Introduction

Probabilistic mixture models are arguably one of the most popular approaches to latent cluster structure discovery from observed data (Redner and Walker 1984; McLachlan and Krishnan 1996; Banerjee et al. 2005c). Naive Bayes (NB) models¹ are a special case of such generative mixture models and have found successful applications in a wide variety of problem domains (Nigam et al. 2000; Domingos and Pazzani 1997; Ng and Jordan 2001). 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 an important restriction that limits their modeling capabilities: they do not allow each data point to belong to different components with varying degrees, i.e., they do not allow mixed memberships. In a recommendation system scenario, such an assumption may indicate that each user only likes one type of movies. In text mining, such an assumption implies that a document can be on only one topic. In reality, the assumption is clearly not true, and becomes a restriction of the mixture models' modeling capability. There are a few existing approaches to relax this assumption to mixed membership, most prominently including multi-cause models (Saund 1994; Ghahramani 1995; Shahami et al. 1997), overlapping mixture models (Banerjee et al. 2005b; Segal et al. 2003; Fu and Banerjee 2008), and aspect models (Hoffman 1999) as well as its generalization—latent Dirichlet allocation (LDA) (Blei et al. 2003; Griffiths and Steyvers 2004). Such mixed-membership (MM) 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 (Blei et al. 2003; Airoldi et al. 2008). However, most of such mixed-membership models only work with a specific type of data (Blei and Jordan 2003) such as text or real valued features, but have not been systematically generalized to deal with arbitrary data types or heterogeneous feature vectors (e.g., user's personal information in online shopping systems, such as age, occupation, monthly expense, etc.), where NB models are still the methods of choice (Mitchell et al. 2004; Yousef et al. 2007). Meanwhile, for most of such mixed-membership models, especially for those with multiple layers of hierarchical structure like LDA, learning the model through a direct application of expectation maximization (EM) (Dempster et al. 1977) algorithm is usually intractable. Two most popular types of approaches to address the problem are variational approximation (Jaakkola 2000; Blei et al. 2003) and Gibbs sampling (Geman and Geman 1984; Griffiths and Steyvers 2004). Unfortunately, most of these existing algorithms are computationally expensive, which restricts the model's wide application to large datasets in real-life cases.

In this paper, we introduce a family of generative models which allows mixed-membership clusterings, while almost maintaining the simplicity of NB models. In

¹ NB in the mixture models setting is an unsupervised clustering algorithm, as opposed to a supervised classification algorithm.

particular, we introduce a family of mixed-membership naive Bayes (MMNB) models, effectively by taking the best of both NB models and mixed-membership topic models such as LDA. MMNB models are significantly more flexible than NB models by using a Dirichlet-discrete prior, while inheriting NB's advantage to deal with heterogenous data. We propose two variational inference algorithms for MMNB, as well as corresponding variational EM algorithms to learn the parameters for any regular exponential family distributions (Banerjee et al. 2005c; Barndorff-Nielsen 1978). The first inference algorithm is based on the ideas originally proposed in the context of LDA (Blei et al. 2003). The second algorithm uses a substantially smaller number of variational parameters, with no dependency on the dimensionality of the dataset, and an application of the same idea in the context of topic modeling gives a new Fast LDA algorithm. By design, the new algorithm has substantially smaller memory requirements, and is orders of magnitudes faster, where the speedup times roughly increases with the dimensionality of data, i.e., the higher dimension the data has, the more computational achievements the algorithm gains compared to the one used in LDA.

Mixed-membership models² achieve good performance in clustering: they yield high clustering accuracy, and also generate mixed-membership vectors which serve as a succinct and interpretable representation of otherwise large and high dimensional data points. However, one important restriction of most existing mixed-membership models is that they are unsupervised models and cannot leverage class label information for classification. Meanwhile, most popular classification algorithms, such as support vector machines (SVM) (Burgess 1998) and logistic regression (LR) (Pampel 2000), perform well on classification, but the classifier itself is often hard to interpret. Therefore, in this paper, we propose discriminative mixed-membership (DMM) models,³ which is a combination of mixed-membership models and logistic regression. In particular, we propose discriminative MMNB (DMMNB) and discriminative LDA (DLDA) as discriminative classification algorithms leveraging mixed-membership models for interpretability, where the mixed-membership vector forms the input to logistic regression for classification.

The effectiveness of MM and DMM models are established through extensive experiments on various types of datasets. We first present results for unsupervised MM models. We show that MMNB models outperform 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. For supervised DMM models, the results show that they achieve higher accuracy than unsupervised MM models, as well as higher/competitive performance compared to the state-of-the-art classification algorithms. More importantly, the new variational inference algorithm used in both MM and DMM is shown to be orders of magnitudes faster than the one used in LDA. In our experiments, we achieve dozens to hundreds of times speedup for Fast LDA/DLDA and 5–10 times speedup for Fast MMNB/DMMNB, with no noticeable loss in the unsupervised setting and even higher accuracy in the supervised setting.

² In this paper, mixed-membership models particularly refer to MMNB and LDA.

³ In this paper, discriminative mixed-membership models particularly refer to discriminative MMNB and LDA.

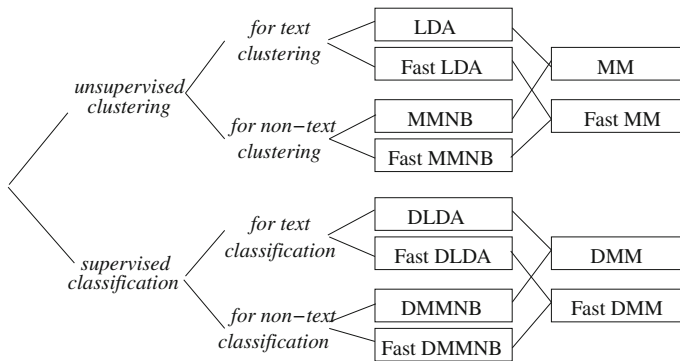


Fig. 1 An overview of the models

Since we propose several different models and their variants, to give a clear overview of these models, we show them in Fig. 1. Overall, we propose two types of models, one is the unsupervised clustering models, the other is the supervised classification models. For clustering, we have LDA and Fast LDA for text clustering, as well as MMNB and Fast MMNB for non-text clustering. LDA and MMNB together are referred to as MM models, and Fast LDA and Fast MMNB together are referred to as Fast MM models. Similarly, for classification, we have DLDA and Fast DLDA for text classification, as well as DMMNB and Fast DMMNB for non-text classification. DLDA and DMMNB together are referred to as DMM models, and Fast DLDA and Fast DMMNB together are referred to as Fast DMM models. All models are new except LDA. The acronym list of the model names is in Table 1.

The rest of the paper is organized as follows: Sect. 2 gives a brief overview on generative mixture models as a background knowledge. Section 3 proposes mixed-membership naive Bayes models. The algorithms of variational inference are presented in

Table 1 The acronym list of model names

Acronym	Full name
LDA	Latent Dirichlet allocation
Fast LDA	Fast latent Dirichlet allocation
MMNB	Mixed-membership naive Bayes
Fast MMNB	Fast mixed-membership naive Bayes
MM	Mixed-membership models
Fast MM	Fast mixed-membership models
DLDA	Discriminative latent Dirichlet allocation
Fast DLDA	Fast discriminative latent Dirichlet allocation
DMMNB	Discriminative mixed-membership naive Bayes
Fast DMMNB	Fast discriminative mixed-membership naive Bayes
DMM	Discriminative mixed-membership models
Fast DMM	Fast discriminative mixed-membership models

Sects. 4 and 5, with Sect. 4 giving the variational inference as a direct generalization of that in LDA, and Sect. 5 giving a fast variational inference. Section 6 proposes the discriminative mixed-membership models. Extensive experimental results are presented in Sect. 7. We review the related literature in Sect. 8 and conclude in Sect. 9.

2 Back ground—generative mixture models

In this section, we give a brief overview of the existing literature on mixture models, which is a background for mixed-membership and discriminative mixed-membership models.

2.1 Finite mixture models

Finite mixture (FM) models are arguably the most widely studied and used form of mixture models (Redner and Walker 1984; Banerjee et al. 2005c). 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 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:

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

where $\Theta = \{\theta_c, [c]_1^k\}$ ($[c]_1^k \equiv c = 1, \dots, k$) are the groups of parameters for the component distributions $\{p(\mathbf{x}|\theta_c), [c]_1^k\}$. Most of the existing literature has focussed on the case where the component distributions belong to a regular exponential family (Banerjee et al. 2005c; Barndorff-Nielsen 1978), and the most widely used mixture models are Gaussian mixture models, where each $p(\mathbf{x}|\theta_c)$ is a Gaussian distribution.

2.2 Naive Bayes models

Naive Bayes (NB) models (Fig. 2a) are a special case of FM models. NB assumes that features of a data point are conditionally independent given the latent component. In particular, with an appropriate univariate exponential family (Banerjee et al. 2005c; Barndorff-Nielsen 1978) 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

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

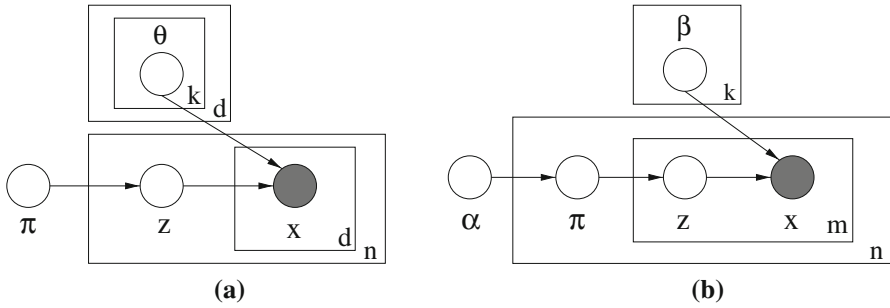


Fig. 2 Graphical model representation of naive Bayes models and latent Dirichlet allocation. **a** Naive Bayes models. **b** Latent Dirichlet allocation

where $\psi_j(\cdot)$ is the cumulant or the log-partition function, and $p_j(x_j)$ is a non-negative base measure. $\psi_j(\cdot)$ determines the exponential family model appropriate for feature j , e.g., Gaussian, Poisson, Bernoulli, etc., and θ_{jc} is the natural parameter corresponding to feature j and component c . Given the prior discrete distribution π over the components, the marginal probability of \mathbf{x} according to naive Bayes is given by

$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^k p(z = c|\pi) \prod_{j=1}^d p\psi_j(x_j|\theta_{jc}). \tag{2}$$

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 flexibility of NB models. Latent Dirichlet allocation (LDA) (Blei et al. 2003; Griffiths and Steyvers 2004) is an elegant extension of standard mixture models by relaxing this assumption in the context of topic modeling, where each data point is a collection of tokens, e.g., a document with a collection of words. LDA assumes that each word in a document potentially comes from a separate topic z , which is generated from a discrete distribution $\text{discrete}(\pi)$ of this document, and all documents share a k -dimensional Dirichlet prior α . The generative process for each document \mathbf{x} is as follows (Fig. 2b):

1. Choose a mixed-membership vector $\pi \sim \text{Dirichlet}(\alpha)$.
2. For each of m words (tokens) $(x_j, [j]_1^m)$ in \mathbf{x} :
 - (a) Choose a topic (component) $z_j = c \sim \text{discrete}(\pi)$.
 - (b) Choose x_j from $p(x_j|\beta_c)$.

$\beta = \{\beta_c, [c]_1^k\}$ is a collection of parameters for k component distributions, with each of them 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

(de Finetti 1990), if the joint distribution of a set of random variables is invariant to permutation, 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 π for the discrete distribution, which is drawn from the Dirichlet distribution $\text{Dirichlet}(\alpha)$. The density function of a document \mathbf{x} is given by

$$p(\mathbf{x}|\alpha, \beta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^m \sum_{c=1}^k p(z_j = c|\pi) p(x_j|\beta_c) \right) d\pi. \quad (3)$$

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 (Jaakkola 2000; Blei et al. 2003) and Gibbs sampling (Geman and Geman 1984; Griffiths and Steyvers 2004).

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 (MMNB) models by taking the best of both worlds.

A “data point” in LDA (Blei et al. 2003) is a collection of tokens, each of which is assumed to be generated from one of the discrete component distributions. The tokens represent the same type of objects, e.g., in case of LDA, all tokens are words. The set of distributions remain 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. Features may be heterogeneous. By “heterogeneous”, we mean the feature vector containing 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.. Using a homogeneous component distribution, LDA is not directly applicable to such heterogeneous features.

As for NB models, while they have been widely used due to their simplicity, and can handle heterogeneous features with measured 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 missing values.
2. Unlike LDA, NB models are not mixed-membership models because they assume that all the features in a feature vector come from the same mixture component.

Such a mixture of unigrams approach (Blei et al. 2003) 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 models where the model itself takes into consideration the sparsity structure of the data points. For a d -dimensional feature vector \mathbf{x} with only a subset of m ($m \leq d$) non-missing features, the density function is given by

$$p(\mathbf{x}|\pi, \Theta) = \sum_{c=1}^k p(z = c|\pi) \prod_{\substack{j=1 \\ \exists x_j}}^d p\psi_j(x_j|\theta_{jc}), \tag{4}$$

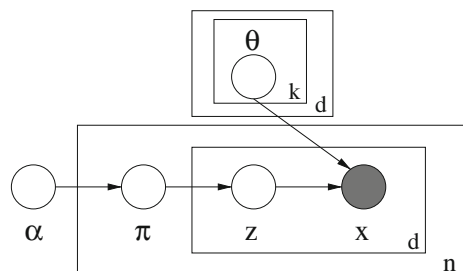
where $\exists x_j$ denotes any observed feature j for \mathbf{x} . Note that the observed feature sets will be potentially different for different \mathbf{x} . Theoretically, the model simply marginalizes over all possible values of the missing features. Operationally, the model is only built over the features whose values are observed, e.g., the movies that have been rated by a certain user.

By focusing only on the observed features, marginal NB 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 of LDA in the context of marginal NB, and propose *mixed-membership naive Bayes* models. In particular, we allow each observed feature x_j of a data point to potentially come from a separate component z_j , which has a Dirichlet-discrete prior on top as in LDA. Given z_j , the generation of each feature still follows marginal NB, which allows MMNB to handle heterogenous features with various types of measured values, so the two limitations of LDA are conveniently addressed. Overall, as a combination of LDA and marginal NB, MMNB takes the best of these two to overcome the limitations of each other.

The graphical model for MMNB is given in Fig. 3. The generative process for \mathbf{x} following MMNB can be described as follows:

1. Choose a mixed-membership vector $\pi \sim \text{Dirichlet}(\alpha)$.
2. For each non-missing feature x_j of \mathbf{x} :
 - (a) Choose a component $z_j = c \sim \text{discrete}(\pi)$.

Fig. 3 Graphical model representation of mixed-membership naive Bayes models



- (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 . We define $\Theta = \{\theta_{jc}, [j]_1^d, [c]_1^k\}$

To make the model fully generative, we also need to generate the sparsity structure of the dataset. In principle, we can assume a fixed Bernoulli distribution $\text{Bernoulli}(\lambda)$ for the entire dataset. The draws from $\text{Bernoulli}(\lambda)$ 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.

From the generative model, the density function for \mathbf{x} is given by:

$$p(\mathbf{x}|\alpha, \Theta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_j}}^d \sum_{c=1}^k p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \right) d\pi. \tag{5}$$

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

$$p(\mathcal{X}|\alpha, \Theta) = \prod_{i=1}^n \int_{\pi_i} p(\pi_i|\alpha) \left(\prod_{\substack{j=1 \\ \exists x_{ij}}}^d \sum_{c=1}^k p(z_{ij} = c|\pi_i) p_{\psi_j}(x_{ij}|\theta_{jc}) \right) d\pi_i. \tag{6}$$

In LDA, an atomic event is the generation of a token (word) x_j from a discrete component distribution, determined by z_j . If there are k components, there would be 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 $p_{\psi_j}(x_j|\theta_{jc})$. If there are k components and d features, the total number of component distributions would be $k \times d$, with k distributions for each of d features 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 heterogeneous 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 across all features, MMNB allows different features to have different distributions and parameters.

1. **MMNB-Gaussian:** Such models have Gaussian distributions for each feature, hence are applicable to the data with real-valued features. Given the model parameters α and $\Omega = \{(\mu_{jc}, \sigma_{jc}^2), [j]_1^d, [c]_1^k\}$, the density function is given by:

$$p(\mathbf{x}|\alpha, \Omega) = \int_{\pi} p(\pi|\alpha) \times \left(\prod_{\substack{j=1 \\ \exists x_j}}^d \sum_{c=1}^k p(z_j = c|\pi) \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} \exp\left(-\frac{(x_j - \mu_{jc})^2}{2\sigma_{jc}^2}\right) \right) d\pi. \quad (7)$$

2. **MMNB-Discrete:** Such models have discrete distributions for each feature, hence are applicable to the data with categorical features. Assuming that feature j can take r_j possible values, each feature j and component c then has a discrete distribution $\{p_{jc}(r), [r]_1^{r_j}\}$, where $p_{jc}(r) \geq 0$ and $\sum_{r=1}^{r_j} p_{jc}(r) = 1$.⁴ Given the model parameters α and $\Omega = \{p_{jc}(r), [r]_1^{r_j}, [j]_1^d, [c]_1^k\}$, the density function is given by

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

4 Inference and estimation

For a given dataset $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the learning task in mixed-membership naive Bayes (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 (6) 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 (Neal and Hinton 1998; Banerjee et al. 2004), which is used to calculate the expectation of 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.

⁴ The representation is over-complete (Wainwright and Jordan 2003). 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.

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

$$p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x}) = \frac{p(\pi|\alpha) \prod_{j=1, \exists x_j}^d p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc})}{\int_{\pi} p(\pi|\alpha) \left(\prod_{j=1, \exists x_j}^d \sum_{c=1}^k p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \right) d\pi} \tag{9}$$

has an intractable partition function, which cannot be computed in closed form. Hence, we introduce a tractable family of parameterized distributions $q_1(\pi, \mathbf{z}|\gamma, \phi)$ as an approximation to $p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x})$, where (γ, ϕ) are free variational parameters. In particular, following Blei et al. (2003), we focus on the family (Fig. 4a)

$$q_1(\pi, \mathbf{z}|\gamma, \phi) = q_1(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^d q_1(z_j|\phi_j), \tag{10}$$

where for each data point, γ is a Dirichlet distribution parameter over π and $\phi = \{\phi_j, [j]_1^d, \exists x_j\}$ are parameters for discrete distributions over the latent components z for all non-missing features. Following Jensen’s inequality (Neal and Hinton 1998; Blei et al. 2003) we have

$$\log p(\mathbf{x}|\alpha, \Theta) \geq E_{q_1}[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)] + H(q_1(\pi, \mathbf{z}|\gamma, \phi)), \tag{11}$$

where $H(\cdot)$ denotes the Shannon entropy. Note that (11) gives a family of lower bounds to the true log-likelihood $\log p(\mathbf{x}|\alpha, \Theta)$, parameterized by (γ, ϕ) . If we denote the corresponding lower bound for data point \mathbf{x}_i by $L(\gamma_i, \phi_i; \alpha, \Theta)$, following (11), we have

$$L(\gamma_i, \phi_i; \alpha, \Theta) = E_{q_1}[\log p(\pi_i|\alpha)] + E_{q_1}[\log p(\mathbf{z}_i|\pi_i)] + E_{q_1}[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] - E_{q_1}[\log q_1(\pi_i|\gamma_i)] - E_{q_1}[\log q_1(\mathbf{z}_i|\phi_i)]. \tag{12}$$

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 update equations that iteratively maximize the lower bound:

$$\gamma_{ic} = \alpha_c + \sum_{\substack{j=1 \\ \exists x_{ij}}}^d \phi_{ijc} \tag{13}$$

$$\phi_{ijc} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right)\right) p_{\psi_j}(x_{ij}|\theta_{jc}), [i]_1^n, [j]_1^d, [c]_1^k, \exists x_{ij}, \tag{14}$$

where γ_{ic} is the c th component of the parameter for variational Dirichlet distribution of the i th data point, ϕ_{ijc} is the c th component of the parameter for the variational discrete

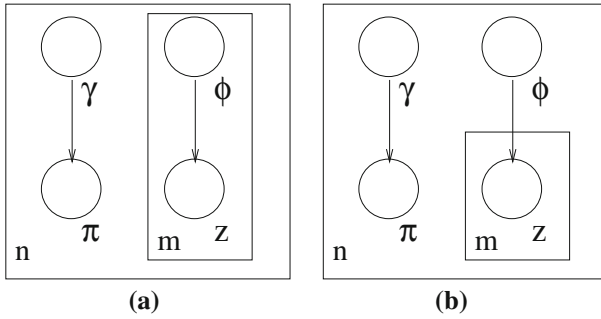


Fig. 4 Variational distributions for MM and Fast MM models. **a** MMNB/LDA. **b** Fast MMNB/LDA

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 Banerjee et al. (2005c), 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 f is the conjugate of the cumulant function Ψ of the family, $b_f = \exp(f(x))p_0(x)$, and $d_f(\cdot, \cdot)$ is the Bregman divergence determined by the function f . Therefore, (14) could be written as

$$\phi_{ijc} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) - d_{f_j}(x_{ij}, \tau_{jc})\right), \tag{15}$$

where τ_{jc} is the mean of the j th feature of the c th component. The above equation shows the following observation: 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 x_{ij} 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 (14). 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 $\log p(\mathcal{X}|\alpha, \Theta)$, i.e., $\sum_{i=1}^n L(\gamma_i^*, \phi_i^*; \alpha, \Theta)$, is a function of the parameters (α, Θ) . Following Redner and Walker (1984); Banerjee et al. (2005c), the parameters Θ can be estimated in 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

$$\tau_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} s_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}}, [j]_1^d, [c]_1^k, \tag{16}$$

where s_{ij} is the sufficient statistic. 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}^2 can be obtained as

$$\mu_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} x_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}} \tag{17}$$

$$\sigma_{jc}^2 = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc} (x_{ij} - \mu_{jc})^2}{\sum_{i=1, \exists x_{ij}}^n \phi_{ijc}}, [j]_1^d, [c]_1^k. \tag{18}$$

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

$$p_{jc}(r) \propto \sum_{i=1}^n \phi_{ijc} \mathbb{1}(x_{ij} = r), [c]_1^k, [j]_1^d, [r]_1^{r_j}, \tag{19}$$

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 possibility of some probability estimates being zero. Such an eventuality does not pose a problem on the training set, but inference on unseen or test 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 test 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 (DeGroot 1970) assuming a Dirichlet prior over each discrete distribution, so that

$$p_{jc}(r) = \sum_{i=1}^n \phi_{ijc} \mathbb{1}(x_{ij} = r) + \epsilon, [c]_1^k, [j]_1^d, [r]_1^{r_j}, \tag{20}$$

for some $\epsilon > 0$.

The update of α is independent of the choice of exponential family distribution. Using Newton–Raphson algorithm (Blei et al. 2003; Minka 2003b) with line search, the updating equation is given by:

$$\alpha'_c = \alpha_c - \eta \frac{g_c - u}{h_c}, [c]_1^k, \tag{21}$$

where

$$g_c = n \left(\Psi \left(\sum_{l=1}^k \alpha_l \right) - \Psi(\alpha_c) \right) + \sum_{i=1}^n \left(\Psi(\gamma_{ic}) - \Psi \left(\sum_{l=1}^k \gamma_{il} \right) \right)$$

$$h_c = -n \Psi'(\alpha_c)$$

$$u = \frac{\sum_{l=1}^k g_l / h_l}{w^{-1} + \sum_{l=1}^k h_l^{-1}}$$

$$w = n \Psi' \left(\sum_{l=1}^k \alpha_l \right).$$

Since α has the constraint of $\alpha_c > 0$, by multiplying the second term of (21) 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 Variational EM for MMNB

Based on the variational inference and parameter estimation updates, it is straightforward to construct a variational EM algorithm to estimate (α, Θ) . Starting with an initial guess $(\alpha^{(0)}, \Theta^{(0)})$, the variational 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 $(t + 1)$ th iteration, we have

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

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 $L(\gamma_i^{(t+1)}, \phi_i^{(t+1)}; \alpha, \Theta)$. Therefore, the objective function is non-decreasing until convergence.

5 Fast variational inference

The variational distribution we have used in Sect. 4 exactly follows the idea proposed for latent Dirichlet allocation (LDA) (Blei et al. 2003), 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 algorithm. Mixed-membership naive Bayes (MMNB) with such a fast variational inference algorithm is referred to as Fast MMNB. We also apply the same idea to LDA and come up with the Fast LDA algorithm. The details of derivation are presented in Appendices A.2 and A.3 for Fast MMNB and Fast LDA respectively.

5.1 Variational approximation

Given the lower bound to log-likelihood of each data point as (11) in Sect. 3, the variational distribution we have used is (10), 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 variational EM algorithm, the optimization is performed 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 (Fig. 4b):

$$q_2(\pi, \mathbf{z}|\phi, \gamma) = q_2(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^d q_2(z_j|\phi). \tag{22}$$

Compared to $q_1(\pi, \mathbf{z}|\phi, \gamma)$ in (10), $q_2(\pi, \mathbf{z}|\phi, \gamma)$ only has one discrete distribution parameter ϕ for each data point. For data points with no missing features, the total

number of ϕ_s decreases from $n \times d$ in (10) to n in (22), accordingly, the number of optimizations over ϕ also decreases from $n \times d$ to n . Such a reduction implies 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 (22), 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

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \tag{23}$$

$$\phi_{ic} \propto \exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right)\right) \left(\prod_{\substack{j=1 \\ \exists x_{ij}}}^d p_{\psi_j}(x_{ij}|\theta_{jc})\right)^{1/m_i}, [i]_1^n, [c]_1^k, \tag{24}$$

where γ_{ic} and ϕ_{ic} are parameters for variational Dirichlet and discrete distributions for the c th component of \mathbf{x}_i respectively. Comparing (24) to (14), in (14), we have the term $p_{\psi_j}(x_{ij}|\theta_{jc})$ in ϕ_{ijc} for each feature j of \mathbf{x}_i , but in (24), since there is only one ϕ_i for all features of \mathbf{x}_i , it contains the geometric mean of $p_{\psi_j}(x_{ij}|\theta_{jc})$ over all non-missing features. γ_{ic} is again independent of the exponential family being used.

5.2 Parameter estimation

After obtaining the variational parameters, we can have 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 Sect. 4 using Newton–Raphson algorithm with line search, and the estimation for Θ has a closed form for exponential family distributions. 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

$$\tau_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ic} s_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ic}}, [j]_1^d, [c]_1^k, \tag{25}$$

where s_{ij} is the sufficient statistic and the natural parameter $\theta_{jc} = \nabla f_j(\tau_{jc})$ by conjugacy, and $f_j(\cdot)$ 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 (25)–(28) are mild variants of (16)–(19) as ϕ_{ic} does not depend on feature j .

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

$$\mu_{jc} = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ic} x_{ij}}{\sum_{i=1, \exists x_{ij}}^n \phi_{ic}} \tag{26}$$

$$\sigma_{jc}^2 = \frac{\sum_{i=1, \exists x_{ij}}^n \phi_{ic} (x_{ij} - \mu_{jc})^2}{\sum_{i=1, \exists x_{ij}}^n \phi_{ic}}, [c]_1^k, [j]_1^d. \tag{27}$$

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

$$p_{jc}(r) \propto \sum_{i=1}^n \phi_{ic} \mathbb{1}(x_{ij} = r) + \epsilon, [c]_1^k, [j]_1^d, [r]_1^{r_j} \tag{28}$$

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, a variational EM algorithm could be constructed to estimate (α, Θ) as in Sect. 4.3.

5.3 Fast LDA

We apply the same idea in Fast MMNB to variational inference in LDA (Blei et al. 2003), yielding Fast LDA. As in Fig. 2b, 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 Blei et al. (2003), the v th word in the dictionary is represented by a V -dimensional vector x such that $x^v = 1$ and $x^u = 0$ for $u \neq v$, and each document \mathbf{x} is represented by m words $\mathbf{x} = \{x_1, x_2, \dots, x_m\}$.

We introduce the same variational distribution for Fast MMNB as in Fig. 4b, i.e., for each document \mathbf{x} , we introduce one Dirichlet distribution parameterized by γ and one discrete distribution parameterized by ϕ . In particular, the variational distribution is given by:

$$q_2(\pi, \mathbf{z}|\phi, \gamma) = q_2(\pi|\gamma) \prod_{j=1}^m q_2(z_j|\phi). \tag{29}$$

The lower bound of the log-likelihood in (3) is again obtained from Jensen’s inequality as in (11). 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{x}_i are as follows:

$$\gamma_{ic} = \alpha_c + m_i \phi_{ic} \tag{30}$$

$$\phi_{ic} \propto \exp \left(\Psi(\gamma_{ic}) - \Psi \left(\sum_{l=1}^k \gamma_{il} \right) + \frac{1}{m_i} \sum_{j=1}^{m_i} \sum_{v=1}^V x_{ij}^v \log \beta_{cv} \right), [i]_1^n, [c]_1^k, \tag{31}$$

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

For fixed values of variational parameters γ and ϕ , maximizing the aggregate lower bound with respect to the model parameters yields the update equation for α and β . In particular, the update equation for α is the same as (21), and the update equation for β is given by:

$$\beta_{cv} \propto \sum_{i=1}^n \left(\phi_{ic} \sum_{j=1}^{m_i} x_{ij}^v \right), [c]_1^k, [v]_1^V. \quad (32)$$

6 Discriminative mixed-membership models

While mixed-membership (MM) models provide a succinct and interpretable representation of otherwise large and high-dimensional datasets, there is an important restriction that they are unsupervised models and cannot leverage class label information for classification. On the other hand, most of the popular classification algorithms, such as support vector machines (SVM) (Burges 1998) and logistic regression (LR) (Pampel 2000) are usually difficult to interpret. Therefore, an accurate discriminative classification algorithm leveraging mixed-membership models for interpretability is highly desirable.

Supervised latent Dirichlet allocation (SLDA) (Blei and McAuliffe 2007) is such a mixed-membership model which takes response variables into account, but it has two limitations preventing it from being used as a classification algorithm:

1. The response variables in SLDA are univariate real numbers assumed to be generated from a normal linear model, whereas the response variables, i.e., labels, are discrete categories in the classification setting. Although the authors pointed out that the response variables can be of various types obtained from generalized linear models, variational inference is difficult in the general case. While a Taylor expansion is recommended (Blei and McAuliffe 2007) to obtain an approximation of the log-likelihood, such an approach forgoes the lower bound guarantee of variational inference.
2. Like latent Dirichlet allocation (LDA), SLDA is designed for text data as a collection of homogeneous tokens. However, most non-text classification tasks, e.g., the UCI benchmark datasets, have features of heterogeneous types with measured values. SLDA is not designed for such data.

In this section, we propose discriminative⁵ mixed-membership (DMM) models as a classification algorithm by combining multi-class logistic regression with unsupervised MM models. In particular, we consider two variants—discriminative latent Dirichlet allocation (DLDA) and discriminative mixed-membership naive Bayes (DMMNB). DLDA is applicable to text classification and uses latent Dirichlet allocation (LDA) (Blei et al. 2003) as the underlying MM model. DMMNB is applicable to non-text classification involving different types (e.g., numerical, categorical) of

⁵ “Discriminative” here does not mean a discriminative model, but a generative model used for classification instead of clustering.

feature vectors and uses mixed-membership naive Bayes (MMNB) as the underlying MM model.

6.1 Discriminative LDA

Assuming there are t classes and k components, the graphical model for DLDA is given in Fig. 5a. It is similar with LDA except that it generates the label y other than the document \mathbf{x} through logistic regression with parameter $\eta = \{\eta_1, \dots, \eta_t\}$, where each η_h for $[h]_1^t$ is a k -dimensional vector and η_t is a zero vector by default. The generative process for each document \mathbf{x} and label y is given as follows:

1. Choose a mixed-membership vector $\pi \sim \text{Dirichlet}(\alpha)$.
2. For each of m words $(x_j, [j]_1^m)$ in the document \mathbf{x} ,
 - (a) Choose a component $z_j = c \sim \text{discrete}(\pi)$.
 - (b) Choose a word $x_j \sim \text{discrete}(\beta_c)$.
3. Choose the label from a multi-class logistic regression $y \sim \text{LR}(\eta_1^T \bar{z}, \eta_2^T \bar{z}, \dots, \eta_t^T \bar{z})$.

\bar{z} is an average of $z_1 \dots z_m$ over all observed words. Note that each z_j in LDA is represented as a k -dimensional unit vector with only the c th entry being 1 if it denotes the c th component. $\text{LR}(\eta_1^T \bar{z}, \eta_2^T \bar{z}, \dots, \eta_t^T \bar{z})$ denotes a logistic transformation on $[\eta_1^T \bar{z}, \eta_2^T \bar{z}, \dots, \eta_t^T \bar{z}]$, which is equivalent to a discrete distribution $(p_1, \dots, p_{t-1}, 1 - \sum_{h=1}^{t-1} p_h)$ with $p_h = \frac{\exp(\eta_h^T \bar{z})}{1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z})}$ for $[h]_1^{t-1}$. In two-class classification, y is 0 or 1 generated from Bernoulli($\frac{1}{1 + \exp(-\eta_1^T \bar{z})}$), i.e., there is only one parameter η_1 to be estimated, η_2 is the zero vector by default.

There are two important properties of DLDA, and of DMM models in general: (1) The k -dimensional mixed membership \bar{z} effectively serves as a low dimensional representation of the original document. While \bar{z} in LDA is inferred in an unsupervised way, it is obtained from a supervised dimensionality reduction in DLDA. (2) DLDA allows the number of classes t and the number of components k in the generative model to be different. If k was forced to be equal to t , for problems with a small number of classes, \bar{z} would have been a rather coarse representation of the document. In particular, for

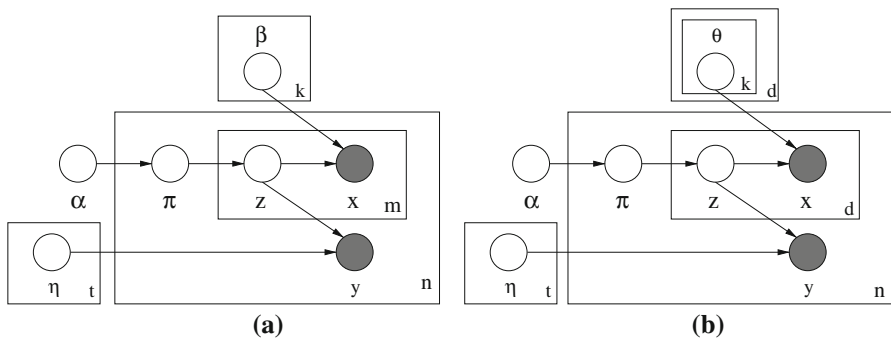


Fig. 5 Graphical models for DLDA and DMMNB. a DLDA. b DMMNB

two-class problems, \bar{z} would lie on the 2-simplex, which may not be an informative representation for classification purposes. Decoupling the choice of k from t prevents such pathologies. In principle, we may find a proper k using Dirichlet process mixture models (Blei and Jordan 2006).

From the generative model, the density function for (\mathbf{x}, y) is given by:

$$p(\mathbf{x}, y|\alpha, \beta, \eta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^m \sum_{c=1}^k p(z_j = c|\pi) p(x_j|\beta_c) \right) p(y|\mathbf{z}, \eta) d\pi. \tag{33}$$

The probability of the entire dataset of n documents and labels $(\mathcal{X} = \{\mathbf{x}_i, [i]_1^n\}, \mathcal{Y} = \{y_i, [i]_1^n\})$ is given by

$$p(\mathcal{X}, \mathcal{Y}|\alpha, \beta, \eta) = \prod_{i=1}^n \int_{\pi_i} p(\pi_i|\alpha) \times \left(\prod_{j=1}^{m_i} \sum_{c=1}^k p(z_{ij} = c|\pi_i) p(x_{ij}|\beta_c) \right) p(y_i|\mathbf{z}_i, \eta) d\pi_i. \tag{34}$$

6.2 Discriminative MMNB

Discriminative MMNB is similar with DLDA except that it is applicable to non-text data and it keeps separate distributions for each feature, as in MMNB. Given the graphical model in Fig. 5b, the generative process for the data point \mathbf{x} and label y is as follows:

1. Choose a mixed-membership vector $\pi \sim \text{Dirichlet}(\alpha)$.
2. For each non-missing feature j in \mathbf{x}
 - (a) Choose a component $z_j = c \sim \text{discrete}(\pi)$.
 - (b) Choose a feature value $x_j \sim p_{\psi_j}(x_j|\theta_{jc})$
3. Choose the label from a multi-class logistic regression $y \sim \text{LR}(\eta_1^T \bar{z}, \eta_2^T \bar{z}, \dots, \eta_t^T \bar{z})$.

Here ψ_j and θ_{jc} jointly decide an exponential family distribution for feature j and component c .

In both DLDA and DMNB, following Blei and McAuliffe (2007), we have used \bar{z} (the mean of z for all words/features) as an input to logistic regression. In principle, any other transformation of z could work, as long as it gives a reasonable representation of the original data point. We choose \bar{z} due to the following two reasons: (1) Optimality: Given a set of data points, their best representative is always the mean according to a wide variety of divergence functions (Banerjee et al. 2005c; Banerjee 2007). We also notice that $\eta_h^T \bar{z} = \eta_h^T E[z] = E[\eta_h^T z]$, which means that if we take the mean of $\eta_h^T z$ on each feature as the input to logistic function, it is equivalent to using $\eta_h^T \bar{z}$ as the input to logistic function. (2) Simplicity. Since z is the latent variable, if

we use other complicated transformation on z such as a non-linear function, it would greatly increase the difficulty in inference and learning.

The density function for (\mathbf{x}, y) is given by

$$p(\mathbf{x}, y|\alpha, \Theta, \eta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^d \sum_{c=1}^k p(z_j = c|\pi) p_{\psi_j}(x_j|\theta_{jc}) \right) p(y|\mathbf{z}, \eta) d\pi. \tag{35}$$

The probability of the entire dataset of n documents and labels $(\mathcal{X} = \{\mathbf{x}_i, [i]_1^n\}, \mathcal{Y} = \{y_i, [i]_1^n\})$ is given by

$$p(\mathcal{X}, \mathcal{Y}|\alpha, \Theta, \eta) = \prod_{i=1}^n \int_{\pi_i} p(\pi_i|\alpha) \times \left(\prod_{j=1}^d \sum_{c=1}^k p(z_{ij} = c|\pi_i) p_{\psi_j}(x_{ij}|\theta_{jc}) \right) p(y_i|\mathbf{z}_i, \eta) d\pi_i. \tag{36}$$

Like MMNB, two special cases of DMMNB are DMMNB-Gaussian and DMMNB-Discrete. The density functions corresponding to (7) and (8) are given by

$$p(\mathbf{x}, y|\alpha, \Omega, \eta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^d \sum_{c=1}^k p(z_j = c|\pi) \times \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} \exp\left(-\frac{(x_j - \mu_{jc})^2}{2\sigma_{jc}^2}\right) \right) p(y|\mathbf{z}, \eta) d\pi \tag{37}$$

for DMMNB-Gaussian and

$$p(\mathbf{x}, y|\alpha, \Omega, \eta) = \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^d \sum_{c=1}^k p(z_j = c|\pi) p_{jc}(x_j) \right) p(y|\mathbf{z}, \eta) d\pi \tag{38}$$

for DMMNB-Discrete.

6.3 Inference and parameter estimation

Since DMM models assume a generative process for both labels as well as the data points, instead of using labels directly to train a classifier, we use both \mathcal{X} and \mathcal{Y} as samples from the generative process to estimate the parameters of DMM models such that the likelihood of observing $(\mathcal{X}, \mathcal{Y})$ is maximized. In particular, we use a same strategy as in Sects. 4 and 5.

For each data point, to obtain a tractable lower bound to $\log p(\mathbf{x}, y|\alpha, \Lambda, \eta)$ ⁶, we introduce a variational distribution $q = q_1$ as in (10) or $q = q_2$ as in (22) to approximate the true posterior distribution $p(\pi, \mathbf{z}|\alpha, \Lambda, \eta)$ over the latent variables. By a direct application of Jensen’s inequality (Blei et al. 2003), the lower bound to $\log p(\mathbf{x}, y|\alpha, \Lambda, \eta)$ is given by:

$$\log p(\mathbf{x}, y|\alpha, \Lambda, \eta) \geq E_q[\log p(\pi, \mathbf{z}, \mathbf{x}, y|\alpha, \Lambda, \eta)] + H(q(\pi, \mathbf{z})). \tag{39}$$

Denoting the lower bound for each data point (\mathbf{x}_i, y_i) with $L(\gamma_i, \phi_i; \alpha, \Lambda, \eta)$, we have

$$\begin{aligned} L(\gamma_i, \phi_i; \alpha, \Lambda, \eta) = & E_q[\log p(\pi_i|\alpha)] + E_q[\log p(\mathbf{z}_i|\pi_i)] + E_q[\log p(\mathbf{x}_i|\mathbf{z}_i, \Lambda)] \tag{40} \\ & - E_q[\log q(\pi_i|\gamma_i)] - E_q[\log q(\mathbf{z}_i|\phi_i)] + E_q[\log p(y_i|\mathbf{z}_i, \eta)]. \end{aligned}$$

As in MM models, using $q = q_1$ yields the regular DMM models and using $q = q_2$ yields Fast DMM models. We will use q to denote q_1 or q_2 without differentiation unless otherwise necessary.

Given the variational distribution q , the first five terms in (40) are exactly the same with the corresponding MM models. The most difficult part is the last term, which cannot be computed exactly even after introducing the variational distribution q , so further approximation is needed. We give the expression for the last term here, and the details of derivation could be found in Appendix A.4. For DLDA, we have

$$\begin{aligned} E_q[\log p(y_i|\mathbf{z}_i, \eta)] \geq & \frac{1}{m_i} \sum_{j=1}^{m_i} \sum_{c=1}^k \phi_{ijc} \left(\sum_{h=1}^{t-1} \eta_{hc} y_{ih} - \frac{1}{\xi_i} \sum_{h=1}^{t-1} \exp(\eta_{hc}) \right) \\ & + \left(1 - \frac{1}{\xi_i} - \log \xi_i \right), \tag{41} \end{aligned}$$

and for DMMNB, we have

$$\begin{aligned} E_q[\log p(y_i|\mathbf{z}_i, \eta)] \geq & \frac{1}{m_i} \sum_{\substack{j=1 \\ \exists x_{ij}}}^d \sum_{c=1}^k \phi_{ijc} \left(\sum_{h=1}^{t-1} \eta_{hc} y_{ih} - \frac{1}{\xi_i} \sum_{h=1}^{t-1} \exp(\eta_{hc}) \right) \\ & + \left(1 - \frac{1}{\xi_i} - \log \xi_i \right), \tag{42} \end{aligned}$$

⁶ Λ denotes β for DLDA and Θ for DMMNB.

Table 2 Updates for variational parameters in DMM and Fast DMM

(a) Updates for ϕ	
DLDA	$\phi_{ijc} \propto \exp \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^k \gamma_{il}) + \sum_{v=1}^V x_{ij}^v \log \beta_{cv} \right. \\ \left. + \frac{1}{m_i} \sum_{h=1}^{t-1} (\eta_{hc} y_{ih} - \exp(\eta_{hc}) / \xi_i) \right)$
Fast DLDA	$\phi_{ic} \propto \exp \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^k \gamma_{il}) + \frac{1}{m_i} \sum_{j=1}^{m_i} \sum_{v=1}^V x_{ij}^v \log \beta_{cv} \right. \\ \left. + \frac{1}{m_i} \sum_{h=1}^{t-1} (\eta_{hc} y_{ih} - \exp(\eta_{hc}) / \xi_i) \right)$
MMNB	$\phi_{ijc} \propto \exp \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^k \gamma_{il}) + \left(-\frac{(x_{ij} - \mu_{jc})^2}{2\sigma_{jc}^2} - \log \sqrt{2\pi\sigma_{jc}^2} \right) \right. \\ \left. + \frac{1}{m_i} \sum_{h=1}^{t-1} (\eta_{hc} y_{ih} - \exp(\eta_{hc}) / \xi_i) \right)$
Fast DMMNB	$\phi_{ic} \propto \exp \left(\Psi(\gamma_{ic}) - \Psi(\sum_{l=1}^k \gamma_{il}) + \frac{1}{m_i} \sum_{j=1, \exists x_{ij}}^d \left(-\frac{(x_{ij} - \mu_{jc})^2}{2\sigma_{jc}^2} - \log \sqrt{2\pi\sigma_{jc}^2} \right) \right. \\ \left. + \frac{1}{m_i} \sum_{h=1}^{t-1} (\eta_{hc} y_{ih} - \exp(\eta_{hc}) / \xi_i) \right)$
(b) Updates for γ	
DLDA	$\gamma_{ic} = \alpha_c + \sum_{j=1}^{m_i} \phi_{ijc}$
Fast DLDA	$\gamma_{ic} = \alpha_c + m_i \phi_{ic}$
DMMNB	$\gamma_{ic} = \alpha_c + \sum_{j=1, \exists x_{ij}}^d \phi_{ijc}$
Fast DMMNB	$\gamma_{ic} = \alpha_c + m_i \phi_{ic}$
(c) Updates for ξ	
DLDA	$\xi_i = 1 + \frac{1}{m_i} \sum_{h=1}^{t-1} \sum_{c=1}^k \sum_{j=1}^{m_i} \phi_{ijc} \exp(\eta_{hc})$
Fast DLDA	$\xi_i = 1 + \sum_{h=1}^{t-1} \sum_{c=1}^k \phi_{ic} \exp(\eta_{hc})$
DMMNB	$\xi_i = 1 + \frac{1}{m_i} \sum_{h=1}^{t-1} \sum_{c=1}^k \sum_{j=1, \exists x_j}^d \phi_{ijc} \exp(\eta_{hc})$
Fast DMMNB	$\xi_i = 1 + \sum_{h=1}^{t-1} \sum_{c=1}^k \phi_{ic} \exp(\eta_{hc})$

where $\xi_i > 0$ is a new introduced variational parameter. Also, for both Fast DLDA and Fast DMMNB, we have

$$E_q[\log p(y_i | \mathbf{z}_i, \eta)] \geq \sum_{c=1}^k \phi_{ic} \left(\sum_{h=1}^{t-1} \eta_{hc} y_{ih} - \frac{1}{\xi_i} \sum_{h=1}^{t-1} \exp(\eta_{hc}) \right) + \left(1 - \frac{1}{\xi_i} - \log \xi_i \right). \tag{43}$$

Maximizing the lower-bound function $L(\gamma_i, \phi_i, \xi_i, \alpha, \Lambda, \eta)$ with respect to the variational parameters gives the update equation of γ_i, ϕ_i and ξ_i as in Table 2. The average of ϕ_{ij} over all existent x_{ij} of \mathbf{x}_i in DMM, or ϕ_i in Fast DMM, gives the posterior of \bar{z} , i.e., the low-dimension representation of each data point. In Table 2, note that the last term in all expressions of ϕ contains y , showing that the low-dimensional representation not only depends on \mathbf{x} , but also depends on y , which means that DMM models achieve supervised dimensionality reduction. Removing the last term gives the expression of ϕ in the corresponding unsupervised settings.

Variational parameters $(\phi_i^*, \gamma_i^*, \xi_i^*)$ from the inference step gives the optimal lower bound to the log-likelihood of (\mathbf{x}_i, y_i) , and maximizing the aggregate lower bound $\sum_{i=1}^n L(\phi_i^*, \gamma_i^*, \xi_i^*, \alpha, \Lambda, \eta)$ over all data points with respect to α, Λ and η respectively yields the estimated parameters. The estimations of α and Λ are the same as in the corresponding MM models. As for η , we have

$$\eta_{hc} = \log \frac{\sum_{i=1}^n \sum_{j=1}^{m_i} y_{ih} \phi_{ijc} / m_i}{\sum_{i=1}^n \sum_{j=1}^{m_i} \phi_{ijc} / (m_i \xi_i)}, [c]_1^k, [h]_1^{t-1}$$

for DMM models, and

$$\eta_{hc} = \log \frac{\sum_{i=1}^n \phi_{ic} y_{ih}}{\sum_{i=1}^n \phi_{ic} / \xi_i}, [c]_1^k, [h]_1^{t-1}$$

for Fast DMM models.

Given the updates for variational and model parameters, a variational EM algorithm could be constructed to optimize the lower bound to the log-likelihood function over variational parameters $(\phi_i, \gamma_i, \xi_i)$ in the E-step, and over the model parameters (α, Λ, η) in the M-step respectively until convergence. The objective function is guaranteed to be non-decreasing.

7 Experimental results

In this section, we present results for clustering using mixed-membership (MM) and for classification using discriminative mixed-membership (DMM) models. First, for MM models, we present three sets of results: (1) comparing mixed-membership naive Bayes (MMNB) with naive Bayes (NB),⁷ (2) comparing Fast MM with MM, and (3) interesting properties in cluster assignments of MMNB. Second, for DMM models, we present three sets of results: (1) comparing (Fast) DMM to corresponding (Fast) MM models. (2) comparing Fast DMM with other state of the art classification algorithms such as support vector machine and logistic regression. (3) the topic lists generated by DLDA.

7.1 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 MMNB and its variants.

UCI datasets: Nine datasets from UCI machine learning repository⁸ are used for our experiments. These datasets are represented as real-valued full matrices without

⁷ In this section, we abuse the terminology by using “naive Bayes models” to refer to the standard NB or marginal NB as appropriate.

⁸ <http://archive.ics.uci.edu/ml/>.

Table 3 The number of data points, features and classes in each UCI dataset

Dataset	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine
Data points	336	214	351	2310	210	208	990	569	178
Features	7	9	32	19	19	60	11	30	13
Classes	8	6	2	7	7	2	11	2	3

missing entries. The numbers of data points, features and classes in each dataset are listed in Table 3.

MovieLens: MovieLens is a movie recommendation dataset created by the GroupLens Research Project.⁹ 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.

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.

Jester: Jester is a joke rating dataset.¹⁰ The original dataset contains 4.1 million continuous ratings of 100 jokes from 73,421 users. The ratings range from -10 to 10 with 10 the best. We pick 1000 users who rate all 100 jokes and use this full data matrix in our experiment.

For the experiments on LDA and its variants, we use 3 text datasets:

Nasa: Nasa is a text dataset downloaded from Aviation Safety Reporting System (ASRS) online database.¹¹ This database contains aviation safety reports submitted by pilots, controllers and others. The dataset used is a subset of the whole database. It contains 4226 documents about the anomalies originating from three sources: flight crew, maintenance, and passengers. The vocabulary size is 604.

Classic3: Classic3 (Dhillon et al. 2003) is a well known text dataset. It contains 3893 documents from three different classes including aeronautics, medicine and information retrieval. The vocabulary size is 5923.

CMU Newsgroup: The CMU Newsgroup is also a benchmark text dataset (Lang 1995). The standard dataset of CMU Newsgroup contains 19,997 messages, collected from 20 different USENET newsgroups. We use three subsets in our experiments: (1) *Diff* is a collection of 3000 messages from 3 different newsgroups with 1000 messages for each class: alt.atheism, rec.sport.baseball and sci.space. The vocabulary size is 7666. (2) *Sim* is a collection of 3000 messages from 3 somewhat similar newsgroups with 1000 messages for each class: talk.politics.guns, talk.politics.mideast, talk.politics.misc. The vocabulary size is 10083. (3) *Same* is a collection of 3000 messages from 3 very similar newsgroups with 1000 messages for each class: comp.graphics, comp.os.ms-windows, comp.windows.x. The vocabulary size is 5932.

⁹ <http://www.grouplens.org/node/73>.

¹⁰ <http://goldberg.berkeley.edu/jester-data/>.

¹¹ http://akama.arc.nasa.gov/ASRSDBOnline/QueryWizard_Begin.aspx.

7.2 Experiments for MM models

In this section, we present results for MM models in three parts. First, we compare MMNB to NB models. Second, we present the results using fast variational inference algorithms. Finally, we show some interesting properties of MM models in cluster assignments.

7.2.1 MMNB vs. NB

In this section, we demonstrate the efficacy of MMNB through the comparison with NB on UCI, Jester, Foodmart and Movielens datasets. We use MMNB-Gaussian for UCI and Jester, MMNB-Poisson for Foodmart, and MMNB-Discrete for Movielens respectively. The results show that MMNB is applicable to different types of data and it achieves a better performance than NB.

Before we make comparison between MMNB and NB, we must note that the parameters of NB effectively has one fewer degree of freedom than MMNB. In particular, the k -dimensional Dirichlet parameter 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, there are k scalars to determine parameter α , but only $k - 1$ scalars 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 lose the ability for generalization and lead to over-fitting on the test set. Therefore, in our experiments, 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 fewer 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 lead to a bad performance on the test set. Thus the results on test sets are more interesting and crucial.

We use *perplexity* as the measurement for comparison. The generative models 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 (Hoffman 1999; Blei et al. 2003) of the entire dataset \mathcal{X} as

$$\text{perplexity}(\mathcal{X}) = \exp \left\{ -\frac{\sum_{i=1}^n \log p(\mathbf{x}_i)}{\sum_{i=1}^n m_i} \right\}, \quad (44)$$

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 (44), 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.

Unless otherwise specified, we use 10-fold cross-validation with random initializations for MM models. 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 once as the test set. We then take the average of results over 10 folds on the training set and the test set respectively. For results on the training set, we train the model on training data by running variational EM as in Sect. 4.3 to obtain the model parameters and variational parameters for calculating the perplexity. 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, and then calculate the perplexity.

The average perplexity of MMNB and NB on UCI, Jester and Foodmart after a 10-fold cross-validation are listed in Table 4. The number of clusters we use for UCI data is the actual number of classes given in the dataset, and the number of clusters for Jester and Foodmart is 10. The p -value is from the paired t-test. It is clear that MMNB has a significantly lower perplexity than NB on most datasets, especially on the more important test-set results, indicating that MMNB fits the data better than NB.

We run more comprehensive experiments on MovieLens. Given a fixed number of classes ($k=20$), Fig. 6 shows the perplexities of MMNB and NB with ϵ varied from 0.01 to 1, where ϵ is the Laplace smoothing parameter as introduced in Sect. 4.2 for MMNB-Discrete case. The overall trend is as follows: when ϵ increases, the perplexity on the training set increases and the perplexity on the 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.

Given 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 of MovieLens are presented as perplexity surfaces in Fig. 7.¹² The key observations are as follows:

1. For the training set results in Fig. 7a, the perplexity surface for MMNB is almost always lower than that of NB over the entire range. NB 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 NB. Both models tend to perform better on the training set with a larger number of latent clusters and a smaller value of the smoothing parameter.
3. For the test set results in Fig. 7b, MMNB achieves a lower perplexity than that of NB for a smaller smoothing parameter. NB performs marginally better than MMNB for high values of the smoothing parameter.
4. The test set performance of MMNB is robust across the entire range of (k, ϵ) , which highlights the stability of the model.

¹² To give a better presentation of the results, the x and y axes do not run in a same direction in (a) and (b).

Table 4 Perplexity of MMNB and NB on training and test sets of UCI, Jester and Foodmart

Dataset	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine	Jester	Foodmart
(a) Training set											
NB	0.4585 ± 0.0061	0.1517 ± 0.0325	1.6574 ± 0.0195	1.8785 ± 0.1641	1.6601 ± 0.1239	0.3073 ± 0.0034	2.1649 ± 0.1470	0.7859 ± 0.0072	4.1454 ± 0.0495	16.5897 ± 0.0535	4.4096 ± 0.0076
MMNB	0.5251 ± 0.0389	0.0675 ± 0.0076	1.4875 ± 0.0434	1.2224 ± 0.0443	1.1485 ± 0.0601	0.2934 ± 0.0027	2.5608 ± 0.0344	0.7800 ± 0.0106	4.1584 ± 0.0567	14.6948 ± 0.1539	4.6218 ± 0.0431
<i>p</i> -value	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	0.0671	<0.05	<0.05
(b) Test set											
NB	0.4993 ± 0.0441	0.1832 ± 0.0651	1.6975 ± 0.1692	1.9041 ± 0.1830	1.8076 ± 0.2812	0.3169 ± 0.0129	2.2413 ± 0.0793	0.8076 ± 0.0792	4.5043 ± 0.4409	17.1023 ± 0.4091	4.6358 ± 0.0464
MMNB	0.6172 ± 0.0724	0.0782 ± 0.0150	1.5268 ± 0.1383	1.2446 ± 0.0430	1.2868 ± 0.1699	0.3004 ± 0.0159	2.7079 ± 0.1279	0.7990 ± 0.0836	4.4978 ± 0.4035	15.1305 ± 0.4001	4.6218 ± 0.0431
<i>p</i> -value	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	0.3967	<0.05	<0.05

MMNB has a lower (better) perplexity on most of the datasets

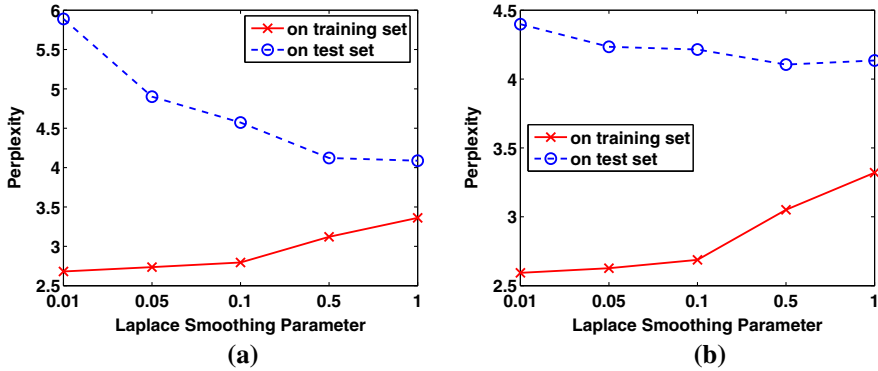


Fig. 6 Perplexities of NB and MMNB with $k = 20$ and varying ϵ on Movielens. With a larger smoothing parameter, perplexity decreases on the training set, and increases on the test set. **a** NB. **b** MMNB

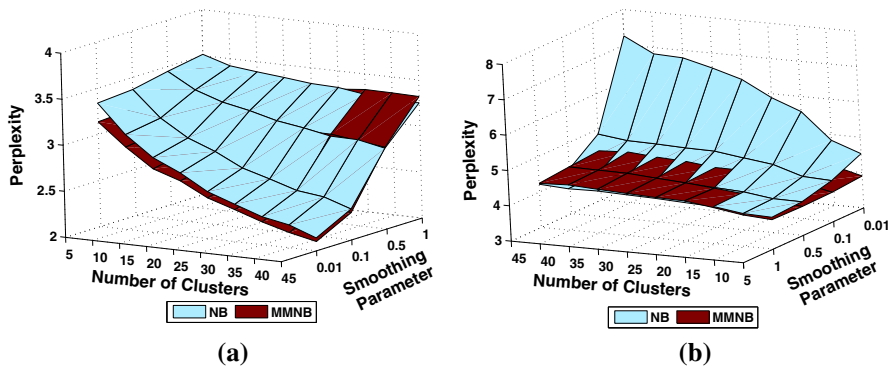


Fig. 7 Perplexity surfaces of NB and MMNB over a range of k and ϵ on Movielens. MMNB mostly has a lower perplexity than NB, and a more stable performance on test set. **a** Training set. **b** Test set

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

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

7.2.2 Fast MM vs. MM

In this section, we demonstrate the advantage of Fast MM compared to the MM in terms of running time and modeling performance measured by perplexity. In addition, for text datasets, we also generate the word lists for topics. The hypothesis is that the Fast MM would achieve a similar performance with MM, but it would be much more computationally efficient.

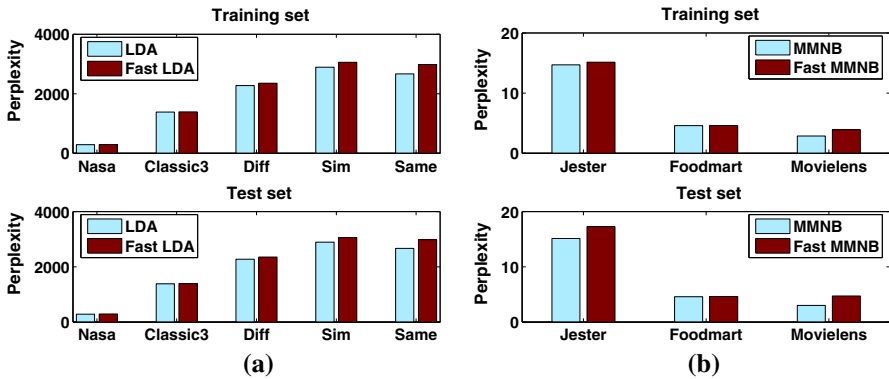


Fig. 8 Perplexity of Fast MM compared to MM. Fast MM achieves similar perplexity with MM. **a** LDA and Fast LDA. **b** MMNB and Fast MMNB

Table 5 Running time (seconds) of Fast MM and MM

(a) LDA and Fast LDA						
	Nasa	Classic3	Diff	Sim	Same	
Dimension	604	5923	7666	10083	5932	
LDA	354.63 ± 1.13	2893.90 ± 4.68	2397.40 ± 28.20	4186.50 ± 80.85	2471.5 ± 243.79	
Fast LDA	2.82 ± 0.07	19.47 ± 0.18	19.77 ± 2.66	34.03 ± 8.37	12.59 ± 3.15	
Speedup times	126	149	121	123	196	
(b) MMNB and Fast MMNB				Jester	Foodmart	Movielens
Dimension	100		1559		1682	
MMNB	215.36 ± 2.02		1190.86 ± 12.92		3664.67 ± 26.73	
Fast MMNB	39.95 ± 2.48		209.15 ± 3.78		356.38 ± 5.01	
Speedup times	5		6		10	

Fast MM is computationally more efficient than MM

We use text datasets for comparing Fast LDA and LDA, and use Jester, Movielens and Foodmart for comparing Fast MMNB and MMNB. The number of clusters on text data is the real number of classes, and the number of clusters on Jester, Movielens and Foodmart is 10. The comparisons of average perplexity and time over 10-fold cross-validation are presented in Fig. 8 and Table 5 respectively. The time shown in the figures 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 similar perplexities for Fast MM and MM on some datasets, and a mildly higher perplexity for Fast MM on some other datasets. The overall performance of these two models are close to each other. As for running time, the results provide the supportive evidence that Fast LDA is 100–200 times faster than LDA, and Fast MMNB is 5–10 times

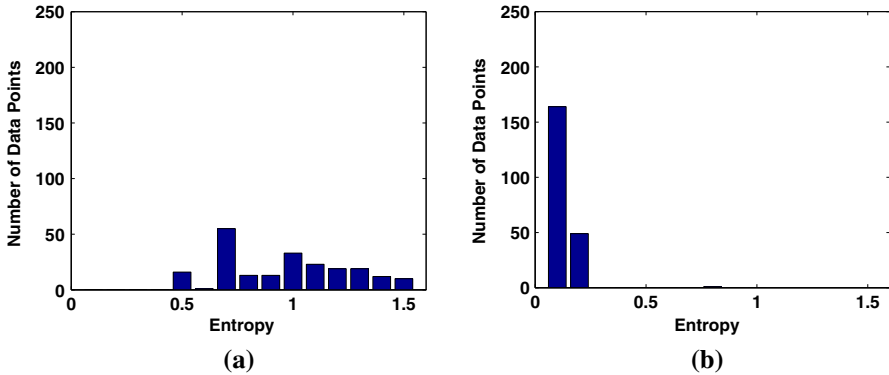


Fig. 9 Histogram of cluster membership entropy on Glass for MMNB and Fast MMNB. **a** MMNB. **b** Fast MMNB

faster than MMNB, which is a significant improvement in computational efficiency. According to the derived update equations, the improvement is directly related to the dimensionality of the data. Since fast variational inference uses one ϕ per data point irrespective of the dimensionality and the regular variational inference uses one ϕ for each dimension of the data point, the speedups achieved by the fast variational inference are more significant in high dimensional data. However, the number of iterations in variational EM algorithm is also an important factor for the running time and it is not determined by the update equations.

We further investigate the cluster assignments of Fast MM. The cluster membership of each data point could be considered as its probability belonging to different clusters. If we calculate the Shannon entropy of the cluster membership, a high entropy indicates a real mixed membership assignment, while a low entropy implies almost a “sole membership”. Figure 9 shows the histograms of cluster membership entropy of MMNB and Fast MMNB on Glass, where each bar denotes the number of data points falling into that range of entropy. While most data points from MMNB have a large entropy over different ranges, the data points from Fast MMNB mostly have a small entropy. Such results also hold for LDA and Fast LDA on text data. The interesting observation indicates that fast variational inference actually generates somewhat “sole membership” while the regular variational inference generates real “mixed membership”.

One possible reason for the sole membership from fast variational inference is as follows: In the E-step, MMNB iterates through (13) and (14), while Fast MMNB iterates through (23) and (24). The expression for γ in (13) contains the summation of ϕ_j over all features j . Since each ϕ_j may take different values, in the sense that each ϕ_j may peak at different component, the summation of ϕ_j may have several peaks on different components. Accordingly, γ will also have several peaks, leading to a mixed membership over those peaked components. In comparison, the expression for γ in (23) has a term of $m\phi$ instead, so no matter which component ϕ peaks at, the peak will be greatly enhanced in γ , and such enhancement in γ will further increases the sole membership nature of ϕ through the term $\exp\left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right)\right)$ in (24).

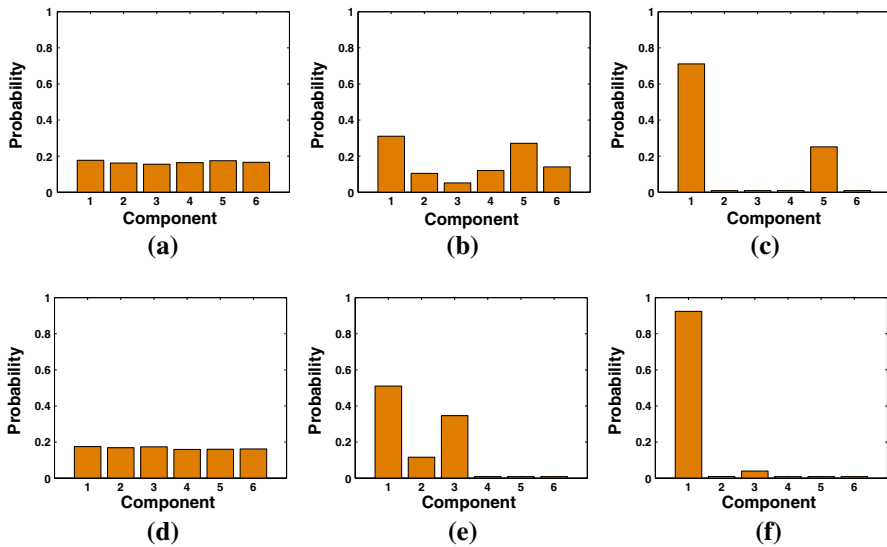


Fig. 10 Posterior over 6 components for one data point in Glass at the beginning, middle and the end of an E-step in MMNB and Fast MMNB. Both algorithms start from an almost uniform distribution, but MMNB ends up with a bimodal distribution showing a mixed membership over two peaked components, and Fast MMNB ends up with a unimodal distribution showing an almost sole membership on the peaked component. Similar results are observed on other datasets. **a** MMNB—beginning. **b** MMNB—middle. **c** MMNB—end. **d** Fast MMNB—beginning. **e** Fast MMNB—middle. **f** Fast MMNB—end

By iterating through γ and ϕ , the accumulated enhancement finally leads to almost a sole membership on the peaked component. Figure 10 shows the posterior of one data point in Glass at different stages (beginning, middle, and the end) of an E-step from MMNB and Fast MMNB, where each bar c shows the probability of the data point belonging to component c among 6 components in total. We can see that at the beginning, both MMNB and Fast MMNB have an almost uniform posterior distribution. As the algorithm runs, the posterior gradually shows some peaks. MMNB finally gives a bimodal distribution (or it could be a multimodal distribution in other examples), and Fast MMNB gives a unimodal distribution. For our experiments, we used a strict stopping criterion for the E-step, however, an early stopping strategy for Fast MMNB may give a mixed membership. The above argument also works for LDA and Fast LDA.

We use 5% of the data as initialization, and run Fast LDA and LDA on the whole data sets to get word lists of topics for text data in Tables 6–10, where the words are listed with decreasing probabilities in each topic. The observations are as follows: First, both LDA and Fast LDA generate appropriate word lists for the topics. For most of the datasets, we can map each list to the given classes. For example, in the result on Nasa, Topic 1 is “flight crew”, Topic 2 is “maintenance”, and Topic 3 is “passenger”. Second, for Nasa, Classic3, and Diff, the datasets with distinct classes of documents, the topic lists generated from Fast LDA and LDA are very similar, even for the rank of words in each topic. For Sim, the dataset with somewhat similar classes

Table 6 Word list for three topics on Nasa

	Topic 1	Topic 2	Topic 3
(a) LDA			
	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			
	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

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”

of documents, Topic 1 and 3 from Fast LDA and LDA are still similar, and there is some difference on Topic 2. The difference is probably because the corresponding class of that topic is “talk.politics.misc”, so it covers several different aspects, which could be extracted in different ways. Despite the difference, the topics generated from Fast LDA and LDA are both qualitatively reasonable/good lists. Finally, for Same, the dataset with very similar classes of documents, the difference between Fast LDA and LDA is more distinct. While we can approximately map three topics from LDA to “comp.windows.x”, “comp.graphics”, and “comp.os.ms-windows” respectively, Fast LDA seems to have comp.graphics in both Topic 2 and 3. Therefore, we believe that LDA performs marginally better than Fast LDA in this case. Given the above observations, we draw the following tentative conclusion in terms of LDA and Fast LDA’s topic modeling performance: If the dataset contains several distinct classes with each document belonging to one, the sole membership generated by Fast LDA is good enough for such datasets, and Fast LDA usually gives very similar topic lists with LDA on such data. When the classes of documents become similar, each document tends to have a mixed membership over different topics, then the sole membership from Fast LDA may not be good enough to extract the topic lists. However, as we can see from

Table 7 Word list for three topics on Classic3

	Topic 1	Topic 2	Topic 3
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) LDA		
	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		
	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	

the examples on Sim and Same, such degeneration of topic modeling performance only happens when the classes are *very* similar or almost the same.

7.2.3 Cluster assignments of MMNB

To obtain a better understanding of MMNB’s behavior, we run more experiments on UCI data to study the relationship between the cluster assignments and modeling performance. Although each data point in the UCI dataset only belongs to one cluster, the cluster assignments from MMNB still conveys interesting information.

The cluster membership entropy indicates the degree of mixed membership. From another perspective, it also shows the model’s degree of confidence when each data point only belongs to one cluster as in UCI. A low entropy implies almost a hard clustering, hence the model’s high confidence of the clustering assignments. A high entropy implies a mixed-membership assignment to multiple clusters, hence the model’s low confidence of the clustering assignments. Therefore, we can learn the relationship between MMNB’s confidence in cluster assignments and its modeling performance. In particular, we use the cluster membership entropy to measure the degree of confidence

Table 8 Word list for three topics on Diff

Topic 1	Topic 2	Topic 3
(a) LDA		
god	space	year
people	earth	game
don	nasa	don
time	launch	team
good	orbit	baseball
religion	system	good
make	shuttle	time
objective	moon	games
point	time	hit
evidence	mission	players
(b) Fast LDA		
god	space	year
people	earth	game
don	nasa	don
religion	launch	team
time	time	baseball
objective	orbit	good
good	system	games
moral	don	time
make	shuttle	hit
point	moon	players

The word lists from LDA and Fast LDA are qualitatively similar. Topic 1 is “alt.atheism”, Topic 2 is “sci.space”, and Topic 3 is “rec.sport.baseball”

in cluster assignments, and use the test-set perplexity to measure the model’s accuracy. The hypothesis is that the more confident the model is on the test set, the higher accuracy it achieves. The experiment runs as follows: we sort all the data points in the test sets in ascending order of their cluster 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 then calculate the perplexities on these five parts separately and draw a perplexity curve. Figure 11 shows the curves as an average of 10-fold cross-validation on 9 UCI datasets. Interestingly, we can see that the perplexity increases monotonically with ascending cluster membership entropy on almost all datasets. Since higher perplexity indicates lower accuracy, and higher cluster membership entropy indicates lower confidence, the observation could be rephrased as: model’s accuracy decreases monotonically with the model’s confidence, i.e., the less confidence the model has, the worse performance it gets. Therefore, the hypothesis is verified. It is a useful property to help us understand the result from MMNB.

Table 9 Word list for three topics on Sim

Topic 1	Topic 2	Topic 3
(a) LDA		
people	people	people
gun	don	israel
don	government	armenian
government	rights	turkish
fbi	men	jews
guns	make	armenians
fire	law	don
law	political	israeli
time	gay	government
batf	free	time
(b) Fast LDA		
people	people	people
gun	president	israel
don	don	armenian
fbi	government	turkish
guns	make	jews
fire	stephanopoulos	armenians
government	states	israeli
koresh	time	jewish
time	state	war
law	health	armenia

The word lists from LDA and Fast LDA are qualitatively similar. Topic 1 is “talk.politics.guns”, Topic 2 is “talk.politics.misc”, and Topic 3 is “talk.politics.mideast”

7.3 Experiments for DMM models

In this subsection, we present experimental results for DMM models in classification. In particular, we compare (Fast) DMM models to (Fast) MM models, and compare Fast DMM to several state-of-the-art classification algorithms with a 10-fold cross validation. For (Fast) DLDA, the experiments are performed on text datasets. For (Fast) DMMNB, the experiments are performed on UCI datasets.

7.3.1 DMM vs. MM

We first compare DMM models to corresponding MM models, using both the regular and fast variational inference for each of them.

For initialization, the model parameters are initialized using all data points and their labels in the training set, in particular, we set the number of components k to be the number of classes t ; use the mean and standard deviation (for Gaussian case only) of the data points in each class to initialize A ; and use n_h/n to initialize each dimension

Table 10 Word list for three topics on Same

Topic 1	Topic 2	Topic 3
(a) LDA		
lib	file	max
expose	image	windows
event	graphics	dos
dpy	program	card
xmu	window	file
libxmu	ftp	bhj
twm	files	win
undefined	jpeg	giz
key	data	run
mydisplay	software	system
(b) Fast LDA		
entry	bit	max
window	card	windows
entries	image	file
program	colour	image
file	graphics	dos
widget	ati	program
rules	ultra	graphics
info	images	files
section	windows	windows
build	conference	don

LDA generates better topic lists. Topic 1 is “comp.windows.x”, Topic 2 is “comp.graphics”, and Topic 3 is “comp.os.ms-windows”

of α , where n_h is the number of data points in class h and n is the total number of data points. For η in DMM we run a cross validation by holding out 10% of training data as the validation set and use the parameters generating the best results on the validation set. In particular, each η_h of $[h]_1^{t-1}$ in η takes value of ru_h , where u_h is a unit vector with the h th dimension being 1 and others being 0, and r takes values from 0 to 100 in steps of 10. In principle, MM models are not used for classification, but given the initialization we have introduced, there is a one-to-one mapping between the component and the class, hence we can measure the accuracy.

The results for DLDA and DMMNB are presented in Tables 11 and 12 respectively. The observations are as follows:

1. On text data, Fast DLDA has a higher accuracy than DLDA. On UCI data, Fast DMMNB generally also has a higher accuracy than DMMNB, with a few exceptions. In Sect. 7.2.2, we have seen that Fast MM achieves a similar performance with MM in clustering, but not as good as it. However, when it comes to classification, Fast DMM has higher accuracy than DMM, making the fast variational inference more advantageous.
2. While DMM models are not necessarily better than MM models, Fast DMM models are almost always better than Fast MM models. Overall, Fast DMM models

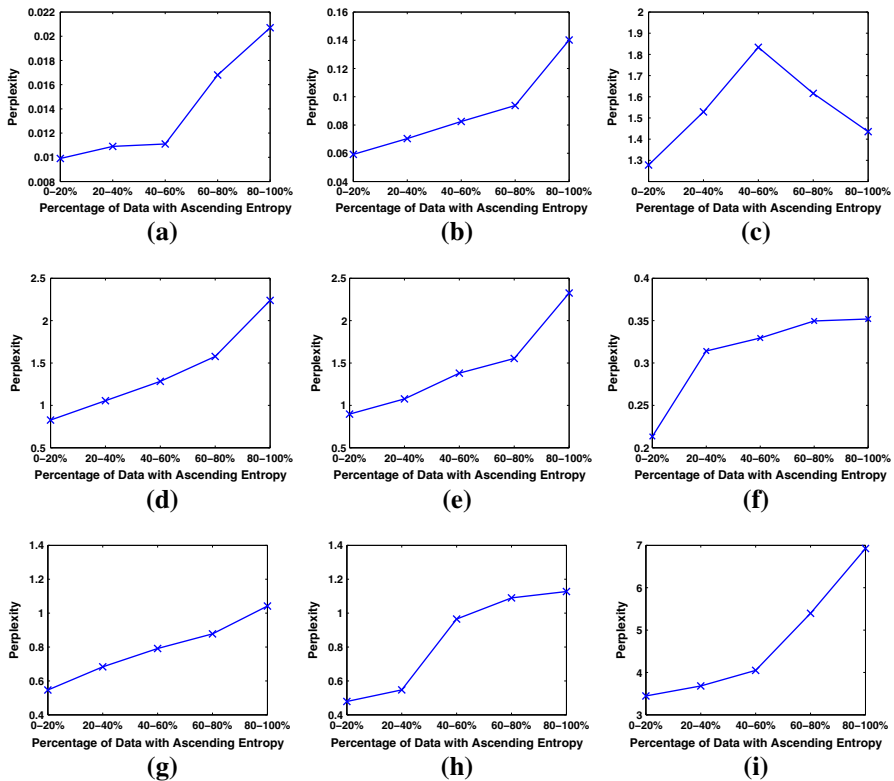


Fig. 11 Perplexities with ascending cluster membership entropy on UCI dataset. Perplexities increase with ascending entropy on most of datasets. **a** Ecoli. **b** Glass. **c** Iono. **d** Seg. **e** Segn. **f** Sona. **g** Vow. **h** Wdbc. **i** Wine

achieve the highest accuracy among four algorithms. The higher accuracy of Fast DMM demonstrates the effects of logistic regression in accommodating label information for DMM models.

As in the unsupervised case, DMM and Fast DMM models generate mixed membership and sole membership respectively. The result of accuracy shows that the sole membership seems to be more helpful than the mixed membership in terms of classification accuracy. The possible reason is that in (single-label) classification scenario, each data point only belongs to one class, hence the sole membership from Fast DMM would probably be more appropriate.

We compare the running time between DMM and Fast DMM. The results for DLDA and DMMNB are presented in Tables 13 and 14 respectively. In Table 14, although most of datasets are small, Fast DMMNB is already faster than DMMNB. Fast DMM's advantage increases when it comes to the larger and higher-dimensional text data as in Table 13, where Fast DLDA is about 20 to 150 times faster than DLDA, showing Fast DMM models' significant superiority in terms of time efficiency, which is consistent with the results in Sect. 7.2.2. Therefore, Fast DMM models are generally more accurate and substantially faster than DMM models.

Table 11 Accuracy for LDA and DLDA ($k=t$) on Text

	Nasa	Classic3	Diff	Sim	Same
LDA	0.9140 ± 0.0140	0.6733 ± 0.0254	0.9677 ± 0.0069	0.8143 ± 0.0161	0.5633 ± 0.0243
DLDA	0.9220 ± 0.0127	0.6710 ± 0.0256	0.9600 ± 0.0089	0.8140 ± 0.0252	0.6267 ± 0.0348
Fast LDA	0.9194 ± 0.0148	0.6748 ± 0.0242	0.9773 ± 0.0110	0.8553 ± 0.0197	0.7730 ± 0.0205
Fast DLDA	0.9237 ± 0.0163	0.6756 ± 0.0234	0.9800 ± 0.0102	0.8653 ± 0.0182	0.7900 ± 0.0315

Fast DLDA has a higher accuracy on all datasets

7.3.2 Fast DMM vs. other classification algorithms

Since Fast DMM models have better performance than DMM models, in this subsection, we use Fast DMM to compare with other classification algorithms. In particular, we compare Fast DMMNB with support vector machine (SVM) (Chang and Lin 2001), logistic regression (LR) and naive Bayes classifier (NBC)¹³ on UCI data; and compare Fast DLDA with SVM, NBC, LR and mixture of von Mises-Fisher (vMF) (Banerjee et al. 2005a) model on text data. Since DMM is a combination of logistic regression and mixed-membership model, we also compare the results from DMM with the results from MM and logistic regression in two steps sequentially.

For Fast DMM models, we run the experiments with an increasing k . In particular, for Fast DMMNB, we use $k = (t, t + 5, t + 10)$, and for Fast DLDA, we use $k = (t, t + 15, t + 30, t + 50, t + 100)$. For initialization of Λ , we use the mean and standard deviation (for Gaussian case only) of the training data in given classes plus some perturbation if $k > t$; for α , we set it to be $1/k$ on each dimension; and for η , we again use a cross validation as in Sect. 7.3.1. For SVM, we use linear and RBF kernel with same cross validation strategy on the penalty parameter and the kernel parameter (for RBF only) taking values from 10^{-5} to 10^5 in multiplicative steps of 10 respectively.

The results for Fast DLDA and DMMNB are presented in Tables 15 and 16. The top parts of the tables are the results from the generative models, and the bottom parts are the results from discriminative classification algorithms. For SVM, we report the highest accuracy of linear and RBF kernels with different parameters. We use bold for the best results among the generative models and use bold and italic for the best results among all algorithms. Three parts of information could be read from the tables:

1. Overall, on text datasets, Fast DLDA does better than all other algorithms, including SVM, on almost all datasets, which is a promising result although more rigorous experiments may be needed to make a further investigation; on UCI datasets, Fast DMMNB also achieves higher accuracy than all other algorithms on most of datasets except SVM, which beats Fast DMMNB five out of nine times.
2. The better performance of Fast DMM models compared to LR on original datasets indicates that the low-dimensional representation from Fast DMM helps the classification.

¹³ Different from Sect. 7.2.1, naive Bayes used in this subsection is the classifier.

Table 12 Accuracy for MMNB and DMMNB ($k=t$) on UCI

	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine
MMNB	0.7895±0.0629	0.6190±0.1052	0.6829 ±0.0579	0.6514 ±0.0293	0.7190 ±0.0853	0.6300 ±0.0789	0.4535±0.0299	0.9321 ±0.0351	0.9606±0.0500
DMMNB	0.7788±0.0554	0.6048±0.1231	0.7314±0.0895	0.6398±0.0397	0.5762±0.1154	0.6102±0.0822	0.6192±0.0571	0.9397±0.0378	0.9647±0.0411
Fast MMNB	0.7950 ±0.0595	0.5952 ±0.0645	0.7486±0.0643	0.6333 ±0.0676	0.5143±0.0834	0.6100±0.0516	0.4969±0.0322	0.9089±0.0309	0.9470±0.0647
Fast DMMNB	0.8152±0.0862	0.5238 ±0.1209	0.8507±0.0891	0.6701±0.0487	0.5286 ±0.1337	0.6600±0.0876	0.6606 ±0.0409	0.9286 ±0.0253	0.9765±0.0304

Fast DMMNB has a higher accuracy on most of the datasets

Table 13 Running time (seconds) of DLDA and Fast DLDA on text data

	Nasa	Classic3	Diff	Sim	Same
Dimension	604	5923	7666	10083	5932
DLDA	549.17 ± 5.74	2176.67 ± 21.62	1752.78 ± 22.36	2344.64 ± 966.50	1981.46 ± 289.2406
Fast DLDA	3.63 ± 0.21	114.34 ± 18.13	27.56 ± 0.61	36.10 ± 2.98	40.18 ± 5.83
Speedup times	151	19	64	65	49

Fast DLDA is computationally more efficient than DLDA

3. Interestingly, for Fast DMMNB, the accuracy increases monotonically with k from t to $t + 10$ on most of the datasets. For Fast DLDA on text data, an increasing of accuracy with a larger k is also observed, although the result goes up and down without a clear trend. One possible reason for the increasing accuracy is as follows: When k is too small, we are performing a drastic dimension reduction to represent each data point in a k -dimensional mixed-membership representation, which may cause a huge loss of information, but the loss may decrease when k increases.

Fast DMM models do dimensionality reduction and classification in one step via a combination of Fast MM and logistic regression. In principle, we may also use these two algorithms sequentially in two steps, i.e., first using Fast MM models to get a low-dimensional representation, and then applying logistic regression on the low-dimensional representation for classification. The results with different choices of k are presented in Tables 17 and 18 for text and UCI data respectively. It is clear that Fast DMM models outperform the Fast MM&LR strategy. Therefore, by combining Fast MM and logistic regression together, Fast DMM achieves supervised dimensionality reduction to obtain a better low-dimensional representation than Fast MM, which helps classification.

7.3.3 Topics from fast DLDA

As we have mentioned, DMM models generate interpretable results. We give an example of several topic word lists on Nasa generated by Fast DLDA ($k = t + 30$) in Table 19. It is also an interesting result demonstrating the effect of allowing a larger number of components than the number of classes ($k > t$), that is, Fast DLDA may discover topics which are not explicitly specified in class labels, while maintaining the predefined number of classes. The first three topics in Table 19 correspond to three classes in Nasa respectively, but topic 4, which we call “passenger medical emergency”, could be considered as a subcategory of the “passenger” class, and it is not specified in the labels. Neither NBC nor SVM is able to generate this type of results.

Table 14 Running time (seconds) of DMMNB and Fast DMMNB

	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine
Dimension	7	9	32	19	19	60	11	30	13
DMMNB	4.65 ± 1.13	2.76 ± 0.49	5.20 ± 3.11	120.26 ± 77.27	4.40 ± 3.55	4.89 ± 4.51	24.82 ± 8.73	3.33 ± 0.40	2.26 ± 0.25
Fast DMMNB	3.97 ± 0.39	2.21 ± 0.21	0.82 ± 0.01	25.37 ± 6.32	1.69 ± 0.63	1.03 ± 0.07	16.18 ± 0.44	1.91 ± 0.11	1.10 ± 0.04
Speedup times	1.17	1.25	6.34	4.74	2.60	4.75	1.53	1.74	2.05

Fast DMMNB is computationally more efficient than DMMNB

Table 15 Accuracy for different classification algorithms on text

	Nasa	Classic3	Diff	Sim	Same
Fast DLDA ($k=t$)	0.9237 ± 0.0163	0.6756 ± 0.0234	0.9800 ± 0.0102	0.8653 ± 0.0182	0.7900 ± 0.0315
Fast DLDA ($k=t+15$)	0.9232 ± 0.0144	0.6858 ± 0.0216	0.9747 ± 0.0121	0.8713 ± 0.0264	0.8458 ± 0.0214
Fast DLDA ($k=t+30$)	0.9301 ± 0.0128	0.6838 ± 0.0234	0.9817 ± 0.0099	0.8707 ± 0.0228	0.8468 ± 0.0190
Fast DLDA ($k=t+50$)	0.9237 ± 0.0138	0.6854 ± 0.0211	0.9823 ± 0.0083	0.8700 ± 0.0230	0.8150 ± 0.0184
Fast DLDA ($k=t+100$)	0.9261 ± 0.0102	0.6866 ± 0.0245	0.9760 ± 0.0108	0.8718 ± 0.0182	0.8347 ± 0.0187
vMF	0.9216 ± 0.0113	0.6509 ± 0.0246	0.9530 ± 0.0071	0.7447 ± 0.0214	0.7600 ± 0.0347
NBC	0.9334 ± 0.0094	0.6766 ± 0.0230	0.9813 ± 0.0069	0.8613 ± 0.0216	0.8410 ± 0.0262
LR	0.9209 ± 0.0157	0.6396 ± 0.0252	0.9553 ± 0.0157	0.6750 ± 0.1330	0.4823 ± 0.1283
SVM	0.9192 ± 0.0146	0.6854 ± 0.0278	0.9563 ± 0.0105	0.8357 ± 0.0156	0.8120 ± 0.2030

Fast DLDA has higher accuracy on most datasets

Table 16 Accuracy for different classification algorithms on UCI

	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine
Fast DMMNB ($k = r$)	0.8152 ± 0.0862	0.5238 ± 0.1209	0.8507 ± 0.0891	0.6701 ± 0.0487	0.5286 ± 0.1337	0.6600 ± 0.0876	0.6606 ± 0.0409	0.9286 ± 0.0253	0.9765 ± 0.0304
Fast DMMNB ($k = r + 5$)	0.8392 ± 0.0836	0.5248 ± 0.0643	0.8543 ± 0.0908	0.7632 ± 0.0412	0.7238 ± 0.1451	0.8100 ± 0.0907	0.6980 ± 0.0267	0.9393 ± 0.0388	0.9882 ± 0.0284
Fast DMMNB ($k = r + 10$)	0.8485 ± 0.0515	0.5667 ± 0.1015	0.8943 ± 0.0786	0.7684 ± 0.0418	0.7624 ± 0.0920	0.8200 ± 0.1509	0.7020 ± 0.0258	0.9375 ± 0.0329	0.9765 ± 0.0411
NBC	0.8363 ± 0.0745	0.4333 ± 0.1318	0.8114 ± 0.0853	0.6850 ± 0.0625	0.6714 ± 0.0910	0.7268 ± 0.0079	0.6737 ± 0.0346	0.9339 ± 0.0266	0.9705 ± 0.0310
LR	0.8030 ± 0.0610	0.5109 ± 0.1234	0.8400 ± 0.0276	0.8307 ± 0.0358	0.7429 ± 0.2048	0.7500 ± 0.0816	0.4515 ± 0.0444	0.9429 ± 0.0250	0.7471 ± 0.1469
SVM	0.8349 ± 0.0670	0.4676 ± 0.0875	0.9171 ± 0.0594	0.9745 ± 0.0096	0.8678 ± 0.0938	0.7450 ± 0.0896	0.8354 ± 0.0469	0.9536 ± 0.0173	0.9765 ± 0.0304

Fast DMMNB has a higher accuracy, except SVM

Table 17 Accuracy on text from A (Fast LDA and logistic regression in two steps) and B (Fast DLDA) with different choices of k

	Nasa	Classic3	Diff	Sim	Same	
$k=t$	A	0.9194±0.0148	0.5609±0.0281	0.9513±0.0268	0.8560±0.0196	0.7733±0.0339
	B	0.9237±0.0163	0.6756±0.0234	0.9800±0.0102	0.8653±0.0182	0.7900±0.0315
$k=t+15$	A	0.9118±0.0124	0.5611±0.0284	0.9756±0.0112	0.8550±0.0226	0.8173±0.0197
	B	0.9232±0.0144	0.6858±0.0216	0.9747±0.0121	0.8713±0.0264	0.8458±0.0214
$k=t+30$	A	0.9080±0.0143	0.5611±0.0284	0.9760±0.0116	0.8530±0.0216	0.8183±0.0168
	B	0.9301±0.0128	0.6838±0.0234	0.9817±0.0099	0.8707±0.0228	0.8468±0.0190
$k=t+50$	A	0.9085±0.0132	0.5596±0.0284	0.9746±0.0123	0.8546±0.0248	0.8040±0.0201
	B	0.9237±0.0138	0.6854±0.0211	0.9823±0.0083	0.8700±0.0230	0.8150±0.0184
$k=t+100$	A	0.8926±0.0942	0.6537±0.0598	0.9423±0.0896	0.7726±0.1715	0.6726±0.6726
	B	0.9261±0.0102	0.6866±0.0245	0.9760±0.0108	0.8718±0.0182	0.8347±0.0187

Fast DLDA achieves higher accuracy, indicating the advantage of supervised dimension reduction

8 Related work

In this section, we present a brief discussion on the existing literatures related to mixed-membership models, including the unsupervised models and supervised models incorporating different types of supervision information.

Probabilistic latent semantic indexing (pLSI) (Hoffman 1999) is an extension of latent semantic indexing (Deerwester et al. 1990). 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. It 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.

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) (Blei et al. 2003) 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, Griffiths and Steyvers (2004) use a full Bayesian model to analyze abstracts from Proceedings of the National Academy of Sciences (PNAS) and gives the mixed-membership of the abstracts belonging to multiple topics. In addition, correlated topic models (Blei and Lafferty 2005) and dynamic topic models (Blei and Lafferty 2006) are able to incorporate the correlation between topics and the evolution of popular topics over years respectively.

Erosheva et al. (2004) generalize LDA by allowing the mixed-membership to be generated from various distributions other than Dirichlet. It could also be applied to multiple types of data points, e.g., papers and citations. However, it is different from

Table 18 Accuracy on UCI from A (Fast MMNB and logistic regression in two steps) and B (Fast DMMNB) with different choices of k

	Ecoli	Glass	Iono	Seg	Segn	Sona	Vow	Wdbc	Wine	
$k = t$	A	0.7666 ± 0.0655	0.4728 ± 0.1184	0.7057 ± 0.1159	0.6082 ± 0.0627	0.4476 ± 0.0680	0.5550 ± 0.0724	0.4121 ± 0.0446	0.9107 ± 0.0303	0.9294 ± 0.1030
	B	0.8152 ± 0.0862	0.5238 ± 0.1209	0.8507 ± 0.0891	0.6701 ± 0.0487	0.5286 ± 0.1337	0.6600 ± 0.0876	0.6606 ± 0.0409	0.9286 ± 0.0253	0.9765 ± 0.0304
$k = t + 5$	A	0.7636 ± 0.0889	0.5061 ± 0.0777	0.8000 ± 0.0819	0.6346 ± 0.0734	0.5026 ± 0.0873	0.6100 ± 0.0699	0.6600 ± 0.0966	0.9142 ± 0.0550	0.9176 ± 0.0794
	B	0.8392 ± 0.0836	0.5248 ± 0.0643	0.8543 ± 0.0908	0.7632 ± 0.0412	0.7238 ± 0.1451	0.8100 ± 0.0907	0.6980 ± 0.0267	0.9393 ± 0.0388	0.9882 ± 0.0284
$k = t + 10$	A	0.8121 ± 0.0619	0.5204 ± 0.0779	0.8600 ± 0.0845	0.6043 ± 0.0931	0.5476 ± 0.1294	0.6450 ± 0.1091	0.4858 ± 0.0455	0.9178 ± 0.0225	0.9235 ± 0.1039
	B	0.8485 ± 0.0515	0.5667 ± 0.1015	0.8943 ± 0.0786	0.7684 ± 0.0418	0.7624 ± 0.0920	0.8200 ± 0.1509	0.7020 ± 0.0258	0.9375 ± 0.0329	0.9765 ± 0.0411

Fast DMMNB achieves higher accuracy, indicating the advantage of supervised dimension reduction

Table 19 Extracted Topics from Nasa dataset using Fast DLDA

Topic 1	Topic 2	Topic 3	Topic 4
runway	maintenance	passenger	passenger
aircraft	aircraft	flight	flight
approach	flight	attendant	medical
tower	minimum equipment list	told	attendant
cleared	time	captain	emergency
landing	check	seat	aircraft
airport	engine	asked	doctor
turn	mechanical	back	landing
taxi	installed	attendants	attendants
traffic	part	aircraft	captain
final	inspection	lavatory	oxygen
controller	work	crew	paramedics

MMNB in that it still assumes that all features are generated from the same component distribution.

[Blei and Jordan \(2003\)](#) propose Gaussian-multinomial LDA (GM-LDA) which is closely related to MMNB. It uses a Gaussian distribution for generating real valued feature vectors of the images, and a discrete distribution for generating the words in image annotation as LDA does. Therefore, GM-LDA is able to handle heterogeneous data with discrete tokens and real valued features, but MMNB is more general than GM-LDA in the sense that it is applicable to arbitrary types of features using exponential family distributions.

Recently, considerable amount of work has been done on mixed-membership of relational data. [Airoldi et al. \(2008\)](#) propose mixed-membership stochastic blockmodels to deal with binary relationships between the objects. [Shan and Banerjee \(2008\)](#) propose Bayesian co-clustering which infers mixed memberships from dyadic data connecting two different entities. The application of the mixed-membership for relational data includes protein-protein interaction analysis ([Airoldi et al. 2008](#)), social network analysis ([Koutsourelakis and Eliassi-Rad 2008](#)), etc..

One of the most recent progresses on mixed-membership models is Bayesian partial membership model (BPM) ([Heller et al. 2008](#)). BPM uses an exponential family distribution for each component cluster, and each data point is modeled as the weighted product over the component distributions, where the weights are the mixed membership over the clusters. Unlike MMNB, BPM does not assume a factorization over the features of a data point. The inference and learning in BPM is based on Markov chain Monte Carlo methods.

Mixed membership models have been extended to supervised learning settings. Supervised LDA (SLDA) ([Blei and McAuliffe 2007](#)) combines LDA with a real-valued response variable. [Flaherty et al. \(2005\)](#) propose labeled latent Dirichlet allocation to incorporate functional annotation of known genes to guide gene clustering. [Fei-Fei and Perona \(2005\)](#) propose a Bayesian model for natural scene categorization.

Lacoste-Julien et al. (2008) propose DiscLDA which determines document position on topic simplex with guidance of labels. Mimno and McCallum (2008) propose a Dirichlet-multinomial regression which accommodates different types of metadata, including labels. Wang et al. (2008) propose a correlated labeling model for multi-label classification. Wang et al. (2009) extend SLDA for image classification and annotation.

Recent years have seen advances in accelerating inference for mixed membership models, especially for LDA. Porteous et al. (2008) propose a fast algorithm for collapsed Gibbs sampling by only checking a subset of topics before drawing a correct sample. Newman et al. (2007) accelerate LDA by doing inference in a distributed way. Our algorithm is different since it maintains one variational distribution per document, which can possibly be used in conjunction with some of the other advances in fast inference.

9 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 Bayesian modeling to allow NB models to generate mixed-memberships for the data points. Blei et al. (2003) 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 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. We also propose discriminative MMNB and LDA, which are supervised mixed-membership classification algorithms by combining multi-class logistic regression with MMNB and LDA respectively. Extensive experiments on a variety of datasets demonstrate that MMNB has a better performance than NB in terms of predictive perplexity and stability. Further, Fast mixed-membership (MM) models exhibit a substantial improvement in computational efficiency compared to MM, with no noticeable loss quantitatively and qualitatively. Finally, Fast DMM models achieve competitive accuracy with the state-of-the-art classification algorithms.

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. In addition, for the inference algorithm, given the algorithm and the results we have, an interesting and natural problem to investigate

would be developing a fast variational inference algorithm while maintaining mixed memberships. Finally, since MMNB is built on NB, it inherits NB’s property that the features are conditionally independent, so we cannot capture the correlation among the features. It would be interesting to develop mixed membership models which can capture feature correlations.

Acknowledgements We want to warmly thank Nikunj Oza for valuable input on discriminative mixed membership models. The research was supported by National Science Foundation grants IIS-0812183, IIS-0916750, National Science Foundation CAREER grant IIS-0953274, and National Aeronautics and Space Administration grant NNX08AC36A.

Appendix A: Variational inference and parameter estimation

In this Appendix, we give derivations for variational inference algorithms. In Appendix A.1, we give the derivation for mixed-membership naive Bayes (MMNB) as a direct generalization of the inference in latent Dirichlet allocation (LDA). In Appendices A.2 and A.3, we give the derivation for Fast MMNB and Fast LDA. The derivation for discriminative mixed-membership (DMM) models are in Appendix A.4.

A.1 MMNB

Given a data point \mathbf{x} , since a direct computation of $\log p(\mathbf{x}|\alpha, \Theta)$ is intractable, following Blei et al. (2003), we introduce for each data point a variational distribution (Fig. 4a)

$$q_1(\pi, \mathbf{z}|\gamma, \phi) = q_1(\pi|\gamma) \prod_{\substack{j=1 \\ \exists x_j}}^d q_1(z_j|\phi_j) \tag{45}$$

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 (Blei et al. 2003), we have:

$$\begin{aligned} \log p(\mathbf{x}|\alpha, \Theta) &= \log \int_{\pi} \sum_{\mathbf{z}} p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta) d\pi \\ &= \log \int_{\pi} \sum_{\mathbf{z}} q_1(\pi, \mathbf{z}|\gamma, \phi) \frac{p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)}{q_1(\pi, \mathbf{z}|\gamma, \phi)} d\pi \\ &\geq \int_{\pi} \sum_{\mathbf{z}} q_1(\pi, \mathbf{z}|\gamma, \phi) \log \frac{p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)}{q_1(\pi, \mathbf{z}|\gamma, \phi)} d\pi \\ &= \int_{\pi} \sum_{\mathbf{z}} q_1(\pi, \mathbf{z}|\gamma, \phi) \log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta) d\pi \end{aligned}$$

$$\begin{aligned}
 & - \int_{\pi} \sum_{\mathbf{z}} q_1(\pi, \mathbf{z}|\gamma, \phi) \log q_1(\pi, \mathbf{z}|\gamma, \phi) d\pi \\
 & = E_{q_1}[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta)] + H(q_1(\pi, \mathbf{z}|\gamma, \phi)). \tag{46}
 \end{aligned}$$

Therefore (46) 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

$$\begin{aligned}
 L(\gamma_i, \phi_i; \alpha, \Theta) & = E_{q_1}[\log p(\pi_i|\alpha)] + E_{q_1}[\log p(\mathbf{z}_i|\pi_i)] + E_{q_1}[\log p(\mathbf{x}_i|\Theta, \mathbf{z}_i)] \\
 & \quad - E_{q_1}[\log q_1(\pi_i|\gamma_i)] - E_{q_1}[\log q_1(\mathbf{z}_i|\phi_i)]. \tag{47}
 \end{aligned}$$

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

$$\begin{aligned}
 E_{q_1}[\log p(\pi_i|\alpha)] & = \log \Gamma\left(\sum_{c=1}^k \alpha_c\right) - \sum_{c=1}^k \log \Gamma(\alpha_c) \\
 & \quad + \sum_{c=1}^k (\alpha_c - 1) \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \\
 E_{q_1}[\log p(\mathbf{z}_i|\pi_i)] & = \sum_{\substack{j=1 \\ \exists x_{ij}}}^d \sum_{c=1}^k \phi_{ijc} \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \\
 E_{q_1}[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] & = \sum_{\substack{j=1 \\ \exists x_{ij}}}^d \sum_{c=1}^k \phi_{ijc} \log p_{\psi_j}(x_{ij}|\theta_{jc}) \\
 E_{q_1}[\log q_1(\pi_i|\gamma_i)] & = \log \Gamma\left(\sum_{c=1}^k \gamma_{ic}\right) - \sum_{c=1}^k \log \Gamma(\gamma_{ic}) \\
 & \quad + \sum_{c=1}^k (\gamma_{ic} - 1) \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \\
 E_{q_1}[\log q_1(\mathbf{z}_i|\phi_i)] & = \sum_{\substack{j=1 \\ \exists x_{ij}}}^d \sum_{c=1}^k \phi_{ijc} \log \phi_{ijc},
 \end{aligned}$$

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 λ_{ij} , yielding

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

Taking derivative with respect to ϕ_{ijc} and setting it to zero, we have

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

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

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

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

In 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 (47) 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 Banerjee et al. (2005c), 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

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

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:

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

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

$$\frac{\partial L}{\partial \alpha_c} = \sum_{i=1}^n \left(\Psi(\gamma_{ic}) - \Psi \left(\sum_{l=1}^k \gamma_{il} \right) \right) + n \left(\Psi \left(\sum_{l=1}^k \alpha_l \right) - \Psi(\alpha_c) \right). \tag{48}$$

The derivation depends on $\{\alpha_{c'}, [c']_1^k, c' \neq c\}$, so there is no closed form solution. Following [Blei et al. \(2003\)](#), we use Newton–Raphson algorithm to update α_c iteratively, where,

$$\frac{\partial L}{\partial \alpha_c \alpha_c} = n \Psi' \left(\sum_{l=1}^k \alpha_l \right) - n \Psi'(\alpha_c) \tag{49}$$

$$\frac{\partial L}{\partial \alpha_c \alpha_{c'}} = n \Psi' \left(\sum_{l=1}^k \alpha_l \right) \quad (c' \neq c), \tag{50}$$

so the Hessian matrix $H(\cdot)$ has (49) on diagonal and (50) 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 (48) and (49, 50) respectively, the update equation for α_c is given by

$$\alpha'_c = \alpha_c - \frac{g_c - u}{h_c}, [c]_1^k, \tag{51}$$

where

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

The problem with the update equation (51) is that it ignores the fact that α has a constraint of $\alpha_c > 0$. Iterating using (51) 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

$$\alpha'_c = \alpha_c - \eta \frac{g_c - u}{h_c}, [c]_1^k. \tag{52}$$

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, (52) becomes (51). 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 (Fig. 4b)

$$q_2(\pi, \mathbf{z} | \gamma, \phi) = q_2(\pi | \gamma) \prod_{\substack{j=1 \\ \exists x_j}}^d q_2(z_j | \phi), \tag{53}$$

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

$$\log p(\mathbf{x} | \alpha, \Theta) \geq E_{q_2}[\log p(\pi, \mathbf{z}, \mathbf{x} | \alpha, \Theta)] - E_{q_2}[\log q_2(\pi, \mathbf{z} | \gamma, \phi)].$$

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

$$L(\gamma_i, \phi_i; \alpha, \Theta) = E_{q_2}[\log p(\pi_i|\alpha)] + E_{q_2}[\log p(\mathbf{z}_i|\pi_i)] + E_{q_2}[\log p(\mathbf{x}_i|\Theta, \mathbf{z}_i)] - E_{q_2}[\log q_2(\pi_i|\gamma_i)] - E_{q_2}[\log q_2(\mathbf{z}_i|\phi_i)] \tag{54}$$

where

$$E_{q_2}[\log p(\pi_i|\alpha)] = \log \Gamma\left(\sum_{c=1}^k \alpha_c\right) - \sum_{c=1}^k \log \Gamma(\alpha_c) + \sum_{c=1}^k (\alpha_c - 1) \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \tag{55}$$

$$E_{q_2}[\log p(\mathbf{z}_i|\pi_i)] = m_i \sum_{c=1}^k \phi_{ic} \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \tag{56}$$

$$E_{q_2}[\log p(\mathbf{x}_i|\mathbf{z}_i, \Theta)] = \sum_{j=1}^d \sum_{c=1}^k \phi_{ic} \log p_{\psi_j}(x_{ij}|\theta_{jc}) \tag{57}$$

$$E_{q_2}[\log q_2(\pi_i|\gamma_i)] = \log \Gamma\left(\sum_{c=1}^k \gamma_{ic}\right) - \sum_{c=1}^k \log \Gamma(\gamma_{ic}) + \sum_{c=1}^k (\gamma_{ic} - 1) \left(\Psi(\gamma_{ic}) - \Psi\left(\sum_{l=1}^k \gamma_{il}\right) \right) \tag{58}$$

$$E_{q_2}[\log q_2(\mathbf{z}_i|\phi_i)] = m_i \sum_{c=1}^k \phi_{ic} \log \phi_{ic}, \tag{59}$$

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:

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

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:

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

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

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

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 to 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 Bregman divergence perspective, the estimation of expectation τ_{jc} is given by

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

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_{jc}^2 and setting them to zero, we get

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

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

For discrete case, we construct the Lagrangian as

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

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

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

The update equation for α is the same with (52).

A.3 Fast LDA

The variational distribution introduced for Fast LDA is the same as (53). Similarly, by applying Jensen’s inequality, the lower bound $L(\phi_i, \gamma_i; \alpha, \beta)$ of the log-likelihood for each document \mathbf{x}_i is given by

$$L(\gamma_i, \phi_i; \alpha, \Theta) = E_{q_2}[\log p(\pi_i|\alpha)] + E_{q_2}[\log p(\mathbf{z}_i|\pi_i)] + E_{q_2}[\log p(\mathbf{x}_i|\beta, \mathbf{z}_i)] - E_{q_2}[\log q_2(\pi_i|\gamma_i)] - E_{q_2}[\log q_2(\mathbf{z}_i|\phi_i)], \tag{60}$$

where the terms 1, 2, 4, 5 are the same with (55), (56), (58) and (59) respectively, and the term 3 could be expanded as:

$$E_{q_2}[\log p(\mathbf{x}_i|\beta, \mathbf{z}_i)] = \sum_{j=1}^{m_i} \sum_{c=1}^k \sum_{v=1}^V \phi_{ic} x_{ij}^v \log \beta_{cv}.$$

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

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

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

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

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

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

Taking derivative with respect to β_{cv} yields

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

The update equation for α is the same with (52).

A.4 DLDA and DMMNB

In this section, we give the derivation for variational inference in Sect. 6.3. Given the lower bound function as (40), the first five terms could easily be obtained following LDA or MMNB depending on which DMM model is used, so we only work on the last term $E_q[\log p(y_i | \mathbf{z}_i, \eta)]$.

The class label y_i is from a multi-class logistic regression $\text{LR}(\eta_1^T \bar{z}, \eta_2^T \bar{z}, \dots, \eta_t^T \bar{z})$, i.e., y_i is from a discrete distribution $[p_1, p_2, \dots, p_{t-1}, 1 - \sum_{h=1}^{t-1} p_h]$ with $p_h = \frac{\exp(\eta_h^T \bar{z})}{1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z})}$, $[h]_1^{t-1}$. Therefore,

$$p(y_i | \mathbf{z}_i, \eta) = \exp \left(\sum_{h=1}^{t-1} \eta_h^T \bar{z}_i y_{ih} - \log \left(1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z}_i) \right) \right).$$

Accordingly,

$$\begin{aligned} & E_q [\log p(y_i | \mathbf{z}_i, \eta)] \\ &= E_q \left[\sum_{h=1}^{t-1} \eta_h^T \bar{z}_i y_{ih} - \log \left(1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z}_i) \right) \right] \\ &= \sum_{h=1}^{t-1} \sum_{c=1}^k \eta_{hc} E_q [\bar{z}_{ic}] y_{ih} - E_q \left[\log \left(1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z}_i) \right) \right]. \end{aligned} \tag{61}$$

The second term of (61) could be expanded as follows:

$$\begin{aligned} & -E_q \left[\log \left(1 + \sum_{h=1}^{t-1} \exp(\eta_h^T \bar{z}_i) \right) \right] \\ & \geq -\log \left(1 + \sum_{h=1}^{t-1} E_q \left[\exp \left(\sum_{c=1}^k \eta_{hc} \bar{z}_{ic} \right) \right] \right) \\ & \geq -\log \left(1 + \sum_{h=1}^{t-1} E_q \left[\sum_{c=1}^k \bar{z}_{ic} \exp(\eta_{hc}) \right] \right) \\ & = -\log \left(1 + \sum_{h=1}^{t-1} \sum_{c=1}^k E_q [\bar{z}_{ic}] \exp(\eta_{hc}) \right) \\ & \geq -\frac{1}{\xi_i} \sum_{h=1}^{t-1} \sum_{c=1}^k E_q [\bar{z}_{ic}] \exp(\eta_{hc}) + 1 - \frac{1}{\xi_i} - \log(\xi_i), \end{aligned} \tag{62}$$

where the first inequality is from Jensen’s inequality, the second inequality is also from Jensen’s inequality noticing that \bar{z}_i is actually a discrete distribution, and the third inequality is from $-\log(x) \geq 1 - \frac{x}{\xi} - \log(\xi)$ (Minka 2003a) by introducing a new variational parameter $\xi > 0$. Given (62),

$$E_q [\log p(y_i | \mathbf{z}_i, \eta)] \geq \sum_{c=1}^k E_q [\bar{z}_{ic}] \sum_{h=1}^{t-1} \left(\eta_{hc} y_{ih} - \frac{1}{\xi_i} \exp(\eta_{hc}) \right) + 1 - \frac{1}{\xi_i} - \log(\xi_i),$$

where in DLDA $E_q[\bar{z}_{ic}] = \frac{1}{m_i} \sum_{j=1}^{m_i} \phi_{ijc}$, in DMMNB, $E_q[\bar{z}_{ic}] = \frac{1}{m_i} \sum_{j=1, \exists x_j}^{m_j} \phi_{ijc}$, and in Fast DLDA/DMMNB, $E_q[\bar{z}_i] = \phi_i$.

Putting $E_q[\log p(y_i | \mathbf{z}_i, \eta)]$ back to (40) gives us the complete expression for $L(\gamma_i, \phi_i; \alpha, \Lambda, \eta)$. By maximizing (40) with respect to the variational and model parameters alternatively, we find the optimal value for model parameters (α, Λ, η) .

References

- Airoldi E, Blei D, Fienberg S, Xing E (2008) Mixed membership stochastic blockmodels. *J Mach Learn Res* 9:1823–1856
- Banerjee A (2007) An analysis of logistic models: exponential family connections and online performance. In: *Proceedings of the 7th SIAM international conference on data mining (SDM)*
- Banerjee A, Dhillon I, Ghosh J, Merugu S (2004) An information theoretic analysis of maximum likelihood mixture estimation for exponential families. In: *Proceedings of the 21st international conference on machine learning (ICML)*
- Banerjee A, Dhillon I, Ghosh J, Sra S (2005a) Clustering on the unit hypersphere using von (M)ises-(F)isher distributions. *J Mach Learn Res* 6:1345–1382
- Banerjee A, Krumpelman C, Basu S, Mooney R, Ghosh J (2005b) Model based overlapping clustering. In: *Proceedings of the 11th international conference on knowledge discovery and data mining (KDD)*, pp 532–537
- Banerjee A, Merugu S, Dhillon I, Ghosh J (2005c) Clustering with Bregman divergences. *J Mach Learn Res* 6:1705–1749
- Barndorff-Nielsen O (1978) *Information and exponential families in statistical theory*. Wiley, Chichester
- Blei D, Jordan M (2003) Modeling annotated data. In: *ACM SIGIR conference on research and development in information retrieval*, pp 127–134
- Blei D, Jordan M (2006) Variational inference for Dirichlet process mixtures. *Bayesian Anal* 1(1):121–144
- Blei D, Lafferty J (2005) Correlated topic models. In: *Proceedings of the 18th annual conference on neural information processing systems (NIPS)*
- Blei D, Lafferty J (2006) Dynamic topic models. In: *Proceedings of the 23rd international conference on machine learning (ICML)*
- Blei D, McAuliffe J (2007) Supervised topic models. In: *Proceedings of the 20th annual conference on neural information processing systems (NIPS)*
- Blei D, Ng A, Jordan M (2003) Latent Dirichlet allocation. *J Mach Learn Res* 3:993–1022
- Burges C (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov* 2:121–167
- Chang C, Lin C (2001) LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- de Finetti B (1990) *Theory of probability*. Wiley, Chichester
- Deerwester S, Dumais S, Landauer T, Furnas G, Harshman R (1990) Indexing by latent semantic analysis. *J Am Soc Inf Sci* 41(6):391–407
- DeGroot M (1970) *Optimal statistical decisions*. McGraw-Hill, New York
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B* 39:1–38
- Dhillon I, Mallela S, Modha D (2003) Information-theoretic co-clustering. In: *Proceedings of the 9th ACM international conference on knowledge discovery and data mining (KDD)*, pp 89–98
- Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. *Mach Learn* 29:103–130
- Erosheva E, Fienberg S, Lafferty J (2004) Mixed-membership models of scientific publications. In: *Proceedings of the national academy of science*, pp 5220–5227
- Fei-Fei L, Perona P (2005) A (B)ayesian hierarchical model for learning natural scene categories. In: *Proceedings of the 15th IEEE international conference of computer vision and pattern recognition (CVPR)*, pp 524–531

- Flaherty P, Gaever G, Jordan M, Arkin A (2005) A latent variable model for chemogenomic profiling. *Bioinformatics* 21:3286–3293
- Fu Q, Banerjee A (2008) Multiplicative mixture models for overlapping clustering. In: Proceedings of the 8th IEEE international conference on data mining (ICDM), pp 791–796
- Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* 6:721–741
- Ghahramani Z (1995) Factorial learning and the EM algorithm. In: Proceedings of the 8th annual conference on neural information processing systems (NIPS)
- Griffiths T, Steyvers M (2004) Finding scientific topics. *Proc Natl Acad Sci USA* 101:5228–5235
- Heller K, Williamson S, Ghahramani Z (2008) Statistical models for partial membership. In: Proceedings of the 25th international conference on machine learning (ICML), pp 392–399
- Hoffman T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 15th conference in uncertainty in artificial intelligence (UAI)
- Jaakkola T (2000) Algorithms for clustering data. MIT Press, Cambridge
- Koutsourelakis P, Eliassi-Rad T (2008) Finding mixed-memberships in social networks. In: Proceedings of the 23rd national conference on artificial intelligence (AAAI)
- Lacoste-Julien S, Sha F, Jordan M (2008) DiscLDA: discriminative learning for dimensionality reduction and classification. In: Proceedings of the 21st annual conference on neural information processing systems (NIPS)
- Lang K (1995) News weeder: Learning to filter netnews. In: Proceedings of the 12th international conference on machine learning (ICML)
- McLachlan G, Krishnan T (1996) The EM algorithm and extensions. Wiley-Interscience, New York
- Mimno D, McCallum A (2008) Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In: Proceedings of the 24th conference in uncertainty in artificial intelligence (UAI)
- Minka T (2003a) A comparison of numerical optimizers for logistic regression. Tech. rep., Carnegie Mellon University
- Minka T (2003b) Estimating a Dirichlet distribution. Tech. rep., Massachusetts Institute of Technology
- Mitchell T, Hutchinson R, Niculescu R, Pereira F, Wang X, Just M, Newman S (2004) Learning to decode cognitive states from brain images. *Mach Learn* 57:145–175
- Neal R, Hinton G (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan M (ed) Learning in graphical models. MIT Press, Cambridge, pp 355–368
- Newman D, Asuncion A, Smyth P, Welling M (2007) Distributed inference for latent Dirichlet allocation. In: Proceedings of the 20th annual conference on neural information processing systems (NIPS)
- Ng A, Jordan M (2001) On discriminative vs generative classifiers: a comparison of logistic regression and naive Bayes. In: Proceedings of the 14th annual conference on neural information processing systems (NIPS)
- Nigam K, McCallum A, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. *Mach Learn* 39(2/3):103–134
- Pampel F (2000) Logistic Regression: A Primer. Sage, Thousand Oaks
- Porteous I, Newman D, Ihler A, Asuncion A, Smyth P, Welling M (2008) Fast collapsed Gibbs sampling for latent Dirichlet allocation. In: Proceeding of the 14th ACM international conference on knowledge discovery and data mining (KDD), pp 569–577
- Redner R, Walker H (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev* 26(2):195–239
- Saund E (1994) Unsupervised learning of mixtures of multiple causes in binary data. In: Proceedings of the 7th annual conference on neural information processing systems (NIPS)
- Segal E, Battle A, Koller D (2003) Decomposing gene expression into cellular processes. In: Proceedings of 8th pacific symposium on biocomputing (PSB)
- Shahami M, Hearst M, Saund E (1997) Applying the multiple cause model to text categorization. In: Proceedings of the 14th international conference on machine learning (ICML), pp 435–443
- Shan H, Banerjee A (2008) Bayesian co-clustering. In: Proceedings of the 8th IEEE international conference on data mining (ICDM), pp 530–539
- Wainwright M, Jordan M (2003) Graphical models, exponential families, and variational inference. Tech. Rep. TR 649, Department of Statistics, University of California at Berkeley
- Wang C, Blei D, Fei-Fei L (2009) Simultaneous image classification and annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)

- Wang H, Huang M, Zhu X (2008) A generative probabilistic model for multi-label classification. In: Proceedings of the 8th IEEE international conference on data mining (ICDM)
- Yousef M, Jung S, Kossenkov A, Showe L, Showe M (2007) Naive Bayes for microRNA target predictions machine learning for microRNA targets. *Bioinformatics* 23(22):2987–2992