Latent Dirichlet Conditional Naive-Bayes Models

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

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

Abstract

In spite of the popularity of probabilistic mixture models for latent structure discovery from data, mixture models do not have a natural mechanism for handling sparsity, where each data point only has a few non-zero observations. In this paper, we introduce conditional naive-Bayes (CNB) models, which generalize naive-Bayes mixture models to naturally handle sparsity by conditioning the model on observed features. Further, we present latent Dirichlet conditional naive-Bayes (LD-CNB) models, which constitute a family of powerful hierarchical Bayesian models for latent structure discovery from sparse data. The proposed family of models are quite general and can work with arbitrary regular exponential family conditional distributions. We present a variational inference based EM algorithm for learning along with special case analyses for Gaussian and discrete distributions. The efficacy of the proposed models are demonstrated by extensive experiments on a wide variety of different datasets.

1. Introduction

Probabilistic mixture models are arguably one of the most popular approaches to latent structure discovery from observed data [12, 7, 2]. Naive-Bayes (NB) models are a special case of such generative mixture models which have found successful applications in a wide variety of domains [11, 4, 10]. In spite of their popularity, mixture models and NB models do not have an explicit mechanism to handle sparse observations. However, several emerging large scale applications generate sparse observations. For example, in a recommender system, a user rates only a very small fraction of all the available movies. An attempt to use a mixture model to discover user clusters based on movie ratings typically leads to unsatisfactory results, as missing entries dominate the data matrix.

In this paper, we introduce conditional naive-Bayes (CNB) models, which generalize NB mixture models to naturally handle sparsity in observed data by conditioning the model on observed features. For the recommender system example, the CNB model is defined over the ratings conditioned on all the movies rated by a user. While CNB models can handle sparsity by design, they inherit another shortcoming of mixture models—all features in a data point are assumed to come from only one component of the mixture. There are a few existing approaches to relax this assumption, most prominently including multi-cause models [13, 14], overlapping mixture models [1], aspect models [6], and latent Dirichlet allocation (LDA) [3, 5]. In LDA, features of a data point are allowed to potentially come from different components of the mixture. For the recommendation system example, the model allows the possibility of a user liking Action, Animation, and Documentaries simultaneously, whereas mixture models will force each user to be consistently of only one type. In addition, LDA allows each data point to have a different prior drawn from a Dirichlet distribution.

Based on the above motivation, we present latent Dirichlet conditional naive-Bayes (LD-CNB) models that are significantly more flexible than mixture models, and can naturally handle sparsity. Following Blei et al. [3], we present a variational approximation method for inference that can work with arbitrary regular exponential family conditional distributions, which include Gaussian and discrete distributions as special cases. We present extensive experimental results for Gaussian and discrete models. One key highlight of our results is that LD-CNB models perform better than CNB models in most settings, and the performance is very stable across a wide range of input parameter choices, even on held out testing sets.

The rest of the paper is organized as follows. In section 2, we present LD-CNB models for exponential family distributions, along with two specific instantiations to Gaussian and discrete conditional distributions. For learning LD-CNB models, a variational approximation based EM algorithm is presented in Section 3. We present experimental results on UCI benchmark and Movielens recommendation system datasets in section 4, and conclude in section 5.

2. Latent Dirichlet Conditional Naive Bayes

In this section, we present latent Dirichlet conditional naive-Bayes (LD-CNB) models. We motivate the model by taking a careful look at settings where standard LDA and NB models have limitations. The proposed LD-CNB is then constructed by taking the best of both the worlds.

A "data point" for LDA [3] is assumed to be a sequence of tokens, each of which is assumed to be generated from a component discrete distribution. The set of distributions remains the same across all tokens, since the tokens are semantically identical, e.g., words in case of LDA [3, 5]. In several applications, there are two important deviations from the above set-up: (i) Instead of a feature being a token, each feature has a measured value, e.g., real, categorical, etc.; and (ii) Different features in the feature set are semantically different. It is highly desirable to be able to extend LDA-style hierarchical models to such settings.

For finite dimensional feature vectors, a naive-Bayes model works well in practice. In particular, the probability of a feature vector $\mathbf{x}$ given the component z is given by

$$p(\mathbf{x}|\theta, z) = \prod_{j=1}^{d} p_{\psi_j}(x_j|z, \theta_j) \ ,$$

where $p_{\psi_j}(x_j|z, \theta_j)$ is the exponential family distribution for feature j, ψ_j determines the appropriate exponential family, and d is the total number of features [2]. This widely used model suffers from two important limitations:
(i) Most large-scale data-sets are sparse, so most feature values x_j will be unknown. For example, in a movie recommendation setting, each user would have rated only a very small fraction of all available movies. The naive-Bayes models have no explicit mechanism to handle sparsity.
(ii) Unlike LDA, the naive-Bayes models assume that all the features x_j corresponding to a feature vector $\mathbf{x}$ come from the same mixture component z. Such a mixture of unigrams approach [3] puts a severe restriction on the modeling power of naive-Bayes.

To address the first draw-back of naive-Bayes models, we introduce Conditional Naive-Bayes (CNB) models, that condition a naive-Bayes model only on a subset of observed features $\mathbf{f} = \{f_1, \ldots, f_m\}$ where $|\mathbf{f}| = m \leq d$. The conditional distribution of $\mathbf{x}$ is given by

$$p(\mathbf{x}|\pi, \Theta, \mathbf{f}) = \sum_{z=1}^{k} p(z|\pi) \prod_{j=1}^{m} p_{\psi}(x_j|z, f_j, \Theta) \ ,$$

where π is the prior distribution over k components, and $\Theta = \{\theta_z, [z]_1^k\}$ ($[z]_1^k \equiv z = 1, \ldots, k$) are the parameters for exponential family distribution. $\psi \equiv \psi_{f_j}$ determines the exponential family model appropriate for feature f_j. Operationally, the model is only over the features $\mathbf{f}$ whose values were observed, e.g., the movies that have been rated by a certain user. To avoid clutter, we have dropped the subscript f_j on ψ. For the movie rating example, $\mathbf{x}$ are the ratings given by a user on the rated movies in $\mathbf{f}$. Note that $\mathbf{f}$ will be potentially different for different users and the model can directly handle such sparsity structures.

To address the second drawback, we use latent Dirichlet allocation on conditional naive-Bayes models. In particular, we assume a Dirichlet prior with parameter α from which the mixing weights π is sampled. Further, for an observed feature f_j, a component z_j is first sampled from π, and x_j is sampled from the corresponding component distribution. Thus, the process of generating a sample $\mathbf{x}$ following LD-CNB model can be described as follows: (i) Choose $\pi \sim \text{Dir}(\alpha)$; (ii) For each of the observed features $f_j, [j]_1^m$: (a) Choose a class $z_j \sim \text{Discrete}(\pi)$, and (b) Choose a feature value $x_j \sim p_{\psi}(x_j|z_j, f_j, \Theta)$. From the generative model, the joint distribution of $(\pi, \mathbf{z}, \mathbf{x})$ is given by

$$p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta, \mathbf{f}) = p(\pi|\alpha) \prod_{j=1}^{m} p(z_j|\pi) p_{\psi}(x_j|z_j, f_j, \Theta) \ .$$

The marginal distribution for a data point $\mathbf{x}$ is obtained by integrating over π and summing over all $\mathbf{z}$. The probability of an entire dataset $X = \{\mathbf{x}_i, [i]_1^n\}$ is given by

$$p(X|\alpha, \Theta, F) = \tag{1}$$
$$\prod_{i=1}^{n} \int_{\pi} p(\pi|\alpha) \left(\prod_{j=1}^{m_i} \sum_{z_{ij}=1}^{k} p(z_{ij}|\pi) p_{\psi}(x_{ij}|z_{ij}, f_{ij}, \Theta) \right) d\pi \ ,$$

where $F = \{\mathbf{f}_i, [i]_1^n\}$ is the set of features. The form of the corpus probability has obvious similarities with that of LDA [3]. There are, however, a few important differences in the detail: (i) the model is conditioned on the observed features, (ii) the model is over the values that the features can take instead of tokens, and (iii) the marginal probability $p_{\psi}(x_{ij}|z_{ij}, f_{ij}, \Theta)$ follows an appropriate exponential family distribution.

In LDA, an atomic event is the generation of a word w_j from a discrete distribution determined by z_j. For a given z_j, the probability of w_j does not depend on j, i.e., where the word occurs in the document. In LD-CNB, an atomic event is the generation of a value x_j for feature f_j from an exponential family distribution $p_{\psi}(x_j|z_j, f_j, \Theta)$. Furthermore, the distribution (family) depends on which feature is being considered, and, may be different for different f_j. Since each feature can have a different family determined by ψ, the LD-CNB model is readily applicable to heterogeneous feature vectors. In addition, LD-CNB can also handle sparse, missing, or variable-length features.

For a concrete exposition to LD-CNB models, we will focus on two specific instantiations of such models based on univariate Gaussian and discrete distributions for each feature in each class:

LD-CNB-Gaussian: Such models are appropriate for real-valued features. Assuming k latent classes and data

dimensionality of d, the model parameters are $\Theta = \{(\mu_{(z,f_j)}, \sigma^2_{(z,f_j)}), [j]_1^d, [z]_1^k\}$, i.e., each feature in each class has a univariate Gaussian distribution.[1] Then, the probability of generating a feature sequence $\mathbf{x}$ from the LD-CNB-Gaussian model is as in (1) with $p_\psi(x_j|z_j, f_j, \Theta) = p(x_j|\mu_{(z_j,f_j)}, \sigma^2_{(z_j,f_j)})$.

LD-CNB-Discrete: Such models are appropriate for categorical features. In general, each feature is allowed to be of a different type and different number of possible values. Assuming k latent classes, d features with r_j possible values for the feature f_j, the model parameters are $\Theta = \{p_{(z,f_j)}(r), [r]_1^{r_j}, [j]_1^d, [z]_1^k\}$ such that for latent class z and feature f_j, $p_{(z,f_j)}$ is a discrete probability distribution over possible values, i.e., $p_{(z,f_j)}(r) \geq 0, [r]_1^{r_j}$ and $\sum_r p_{(z,f_j)}(r) = 1$.

3. Inference and Learning

For a given corpus X, the learning problem for LD-CNB can be posed as estimating (α^*, Θ^*) such that $p(X|\alpha^*, \Theta^*, F)$ is maximized. However, for a given choice of parameters (α, Θ), computing $p(\mathbf{x}|\alpha, \Theta, F)$ directly is intractable. As a result, we pose and solve a surrogate learning problem of choosing (α^*, Θ^*) that maximizes a variational lower bound on $p(\mathbf{x}|\alpha^*, \Theta^*, F)$.

3.1 Variational Inference

For any distribution $q(\pi, \mathbf{z})$ approximating the latent variable distribution $p(\pi, \mathbf{z}|\alpha, \Theta, \mathbf{x}, F)$, we have [9]

$$\log p(\mathbf{x}|\alpha, \Theta, F) \geq E_q[\log p(\pi, \mathbf{z}, \mathbf{x}|\alpha, \Theta, F)] + H(q(\pi, \mathbf{z})), \quad (2)$$

where $H(\cdot)$ denotes the Shannon entropy. In variational approximation, one works with a tractable family of parametric distributions $q(\pi, \mathbf{z}|\gamma, \phi)$, thereby getting a lower bound on $\log p(\mathbf{x}|\alpha, \Theta, F)$ using (2). The best lower bound can be computed by optimizing over the free variational parameters (γ, ϕ). Following Blei et al. [3], we use

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

where γ is a Dirichlet parameter and $\phi = (\phi_1, \ldots, \phi_m)$ are multinomial parameters. Following (2), we can get a correspond lower bound $L(\gamma, \phi; \alpha, \Theta)$ given by

$$L(\gamma, \phi; \alpha, \Theta) = E_q[\log p(\pi|\alpha)] + E_q[\log p(\mathbf{z}|\pi)] + E_q[\log p(\mathbf{x}|\mathbf{z}, \Theta)] + H(q(\pi)) + H(q(\mathbf{z})) \ .$$

The lower bound $L(\gamma, \phi; \alpha, \Theta)$ can be iteratively maximized over the free parameters (γ, ϕ) using the following set of update equations:

$$\phi_{(z_j,f_j)} \propto \exp\left(\Psi(\gamma_{z_j}) - \Psi\left(\sum_{z_{j'}=1}^{k} \gamma_{z_{j'}}\right)\right) p_\psi(x_j|z_j, f_j, \Theta)$$

$$\gamma_{z_j} = \alpha_{z_j} + \sum_{j=1}^{m} \phi_{(z_j,f_j)} \ .$$

For the LD-CNB-Gaussian model and LD-CNB-Discrete models, the appropriate iterations can be obtained by replacing the corresponding distributions in place of $p_\psi(x_j|z_j, f_j, \Theta)$ in the update equation for $\phi_{(z_j,f_j)}$. The form of the updates for γ_{z_j} is independent of the exponential family being used.

3.2 Parameter Estimation

We use the lower bound $L(\gamma, \phi; \alpha, \Theta)$ as a surrogate objective function to be maximized since the original log-likelihood is intractable. Note that for a fixed value of the variational parameters (γ, ϕ), say obtained by variational inference, the lower bound is a function of the parameters (α, Θ). Following [3], the update for α can be computed using a linear time Newton-Raphson iteration. Further, following [12, 2], the parameters Θ can be estimated in closed form for all exponential family distributions.

LD-CNB-Gaussian: For Gaussians, taking derivative with respect to $\mu_{(z_j,f_j)}$ and $\sigma^2_{(z_j,f_j)}$, the exact update equations can be obtained as

$$\mu_{(z_j,f_j)} = \frac{\sum_{i=1}^n \phi_{i(z_j,f_j)} x_{ij}}{\sum_{i=1}^n \phi_{i(z_j,f_j)}}$$

$$\sigma^2_{(z_j,f_j)} = \frac{\sum_{i=1}^n \phi_{i(z_j,f_j)} (x_{ij} - \mu_{(z_j,f_j)})^2}{\sum_{i=1}^n \phi_{i(z_j,f_j)}} \ ,$$

where $\phi_{i(z_j,f_j)}$ is the variational parameter for component z_j of feature f_j of observation $\mathbf{x}_i$.

LD-CNB-Discrete: For a discrete distribution $p_{(z_j,f_j)}$ over $[r]_1^{r_j}$ values for feature f_j, taking derivative with respect to each component $p_{(z_j,f_j)}(r)$, we have $p_{(z_j,f_j)}(r) \propto \sum_{i=1}^n \phi_{i(z_j,f_j)} \mathbf{1}(r|i, f_j)$, where $\mathbf{1}(r|i, f_j)$ is the indicator of observing value r for feature f_j in observation $\mathbf{x}_i$. To avoid 0 probabilities, applying Laplace smoothing for some $\epsilon > 0$, we get

$$p_{(z_j,f_j)}(r) \propto \sum_{i=1}^{n} \phi_{i(z_j,f_j)} \mathbf{1}(r|i, f_j) + \epsilon \ .$$

3.3 EM for LD-CNB

Starting with an initial guess (α_0, Θ_0), the EM algorithm to estimate (α^*, Θ^*) alternates between two steps:

[1] Naive-Bayes for Gaussians has the exact same set Θ of parameters.

	LD-CNB-Gaussian		CNB-Gaussian	
	train	test	train	test
balance	**4.5071**	**3.3082**	5.8241	5.8888
cmc	**0.6557**	**0.7136**	6.1544	6.1915
derm	**0.1814**	**0.1685**	4.5756	4.5735
glass	**0.4370**	**0.5103**	1.5766	1.6081
iono	**1.4519**	**1.5101**	1.6269	1.6520
iris	**1.7762**	**1.9563**	1.8993	2.0090
lung	**0.2449**	**0.4102**	2.2339	2.6071
musk	**1.6880**	**1.7513**	2.6639	2.6836
pima	**0.2280**	**0.2740**	1.1163	1.1314
wine	**1.4426**	**1.5237**	3.1618	3.2646

Table 1. Perplexity of LD-CNB-Gaussian and CNB-Gaussian on training and testing sets.

1. E-Step: Given (α, Θ), for each data point $\mathbf{x}_i$, find the variational parameters (γ_i^*, ϕ_i^*) that maximize $L(\gamma, \phi; \alpha, \Theta, \mathbf{f}_i)$. $L(\gamma_i^*, \phi_i^*; \alpha, \Theta, \mathbf{f}_i)$ gives a lower bound to $\log p(\mathbf{x}_i|\alpha, \Theta, \mathbf{f}_i)$.

2. M-Step: Maximize the aggregate lower bound $\sum_{i=1}^{n} L(\gamma_i^*, \phi_i^*; \alpha, \Theta, \mathbf{f}_i)$ with respect to (α, Θ) in order to obtain an improved parameter estimate.

4. Experimental Results

In this section, we present two sets of experimental results comparing CNB and LD-CNB models. The first set focuses on Gaussian models and uses UCI benchmark datasets. The second set focuses on discrete models and uses the Movielens recommendation dataset.

4.1 Gaussian Models

In this section, we compare modeling performance of CNB-Gaussian with that of LD-CNB-Gaussian. For all of the datasets we consider, all features are available for all instances. As a result, the CNB model becomes exactly equivalent to the NB model based on all features.

Datasets and Methodology: Ten datasets from UCI machine learning repository, as shown in first column of Table 1, are used for our experiments. For evaluation purpose, we compute the *perplexity* [6, 3] of the entire corpus X as:

$$Perplexity(X) = \exp\Big\{ - \frac{\sum_{i=1}^{n} \log p(\mathbf{x}_i)}{\sum_{i=1}^{n} m_i} \Big\}, \qquad (3)$$

where m_i is the number of observed features for $\mathbf{x}_i$. In the case of UCI dataset, m_i is the same for all instances in each dataset. Perplexity is a monotonically decreasing function of log-likelihood, implying that *lower perplexity is better* since the model can explain the data better. Note that the comparison is fair for all practical purposes, since the LD-CNB models use only one additional parameter compared to CNB. We used the same initial values of μ, σ, and α for two models. All results reported on the testing sets are the average of 10-fold cross validation.

Results: The results are presented in Table 1. It is clear that LD-CNB-Gaussian consistently outperforms CNB-Gaussian on all of the ten datasets. Since all features are observable, CNB-Gaussian is equivalent to the corresponding NB-Gaussian. Figure 1 shows some example learning curves of LD-CNB-Gaussian and CNB-Gaussian on iris, lung, and wine datasets. We display only the first few iterations, but the trend shown is maintained till the end. In general, the perplexity drops dramatically in the first few iterations, and then it slows down until convergence.

4.2 Discretes Models

In this section, we compare the modeling performance of CNB-Discrete with that of LD-CNB-Discrete on the Movielens recommendation dataset [8]. Since each user rates only a small fraction of the available movies, the CNB model is different from the NB model in this case. In fact, the dataset provides a typical situation where applying NB directly can be problematic since most features (movie ratings) corresponding to a data point (user) are missing.

Datasets and Methodology: The Movielens dataset is a movie recommendation dataset created by the Grouplens Research Project. The dataset we use consists of 100,000 ratings (1-5) for 1682 movies by 943 users. Since each user only rates a few movies, the data matrix is very sparse.

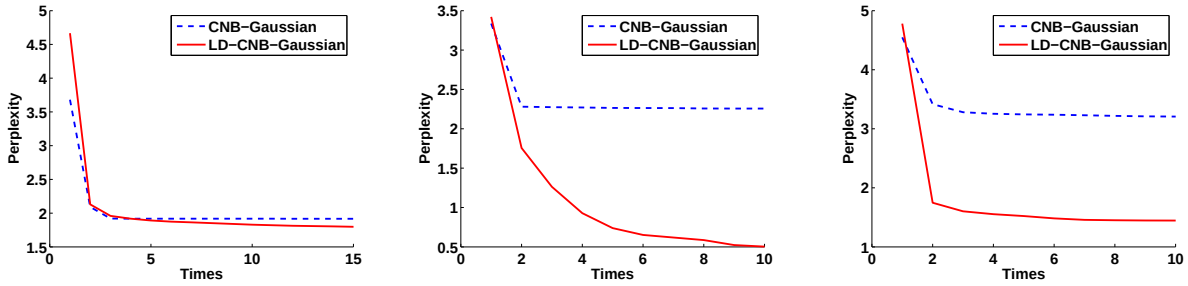
We ran experiments on three models: CNB-Discrete, LD-CNB-Discrete, and a version of NB which treats the missing entries as a sixth category. For CNB-Discrete and LD-CNB-Discrete, since each user has potentially rated a different number of movies, the users are actually represented by a set of variable-length features.

We use the same initialization for CNB-Discrete and LD-CNB-Discrete. For experiments on the training set, we used the entire dataset for both training and testing. For experiments on the testing set, we held out 10% data for testing purpose, and trained on remaining 90%. Further, we applied Laplace smoothing to all three models with a parameter ϵ.

We again use perplexity defined in (3) for evaluation. An important difference with the UCI datasets is that the number of features m_i can be different for different users. Since NB cannot work with variable length sequences, we use *per-user perplexity* to compare CNB and NB:

$$UserPerp(X) = \exp\Big\{ - \frac{\sum_{i=1}^{n} \log p(\mathbf{x}_i)}{n} \Big\}. \qquad (4)$$

Results: For a fixed ϵ (ϵ=0.1) and a fixed number of classes k ($k = 20$), the results are listed in Table 2. LD-CNB-Discrete consistently outperforms CNB-Discrete. Further,



(a) Learning curve on iris (b) Learning curve on lung (c) Learning curve on wine

Figure 1. Learning curve of CNB-Gaussian and LD-CNB-Gaussian over iterations.

	perplexity		per-user perplexity	
	train	test	train	test
NB-Discrete	–	–	1.4309×10^{167}	1.1074×10^{183}
CNB-Discrete	2.7950	4.5731	2.1724×10^{47}	1.2640×10^{70}
LD-CNB-Discrete	**2.6871**	**4.2144**	$\mathbf{3.3297 \times 10^{45}}$	$\mathbf{2.1649 \times 10^{66}}$

Table 2. Perplexity and per-user perplexity of three models on training and testing set while $\epsilon = 0.1$ and $k = 20$.

CNB-Discrete outperforms NB-Discrete in terms of per-user perplexity, as CNB-Discrete is able to concentrate on the meaningful non-zero features while NB-Discrete is dominated by the meaningless zero entries.

We ran extensive experiments for a range of values for k and ϵ. In particular, the number of classes k was varied from 5 to 45 in steps of 5, with ϵ set to 0.01, 0.1, 0.5, and 1. The overall results for entire (k, ϵ) range are presented as perplexity surfaces in Figures 2 and 3 for training and testing sets respectively. For training set results in Figure 2, we note that the perplexity surface for LD-CNB-Discrete is almost always lower than that of CNB-Discrete over the entire range. Both models tend to perform better with a larger k and a smaller ϵ. For testing set results in Figure 3, the perplexity surface for LD-CNB-Discrete is better than that of CNB-Discrete for a smaller ϵ and a smaller k. The testing set performance of LD-CNB-Discrete being very consistent across the entire range of (k, ϵ) is reassuring, and highlights the stability of the model. Overall, LD-CNB-Discrete demonstrates better performance on training set and more consistent and mostly better performance on testing set.

In Figures 2 and 3, if we fix the number of latent classes, say $k = 20$, we could see the perplexity trend with varied values of ϵ. Generally, for larger values of ϵ, the perplexity on training set is higher, and the perplexity on testing set is lower. The result is consistent with the Bayesian intuition behind smoothing. In particular, a lower value of the Laplace smoothing parameter implies a higher confidence on the parameters learnt from the training set. The learnt parameters will surely have a good performance on the training set itself, but not necessarily on the testing set. On the other hand, larger value of the smoothing parameter implies a conservative approach, which need not do as well on training set, but may perform reasonably well on testing set, especially if the training set is noisy or sparse. Therefore, we observed the ideal behavior one would expect as an effect of smoothing.

In order to gain a better understanding, we ran several related experiments to test for failure modes of LD-CNB-Discrete model. In the experiments on a small subset of the Movielens dataset, consisting of the ratings of 50 users on a total of 1084 movies, CNB mostly outperformed LD-CNB across a range of values of (k, ϵ), especially when ϵ is large.

We also compared per-user perplexity of NB-Discrete and CNB-Discrete on the full Movielens dataset. The results are shown in Figures 4 and 5. Interestingly, the per-user perplexities do not change much with the number of latent classes. Further, the perplexity for CNB-Discrete is consistently better than NB-Discrete. The poor performance of NB-Discrete demonstrates the risk of NB modeling with missing feature values. The improvement due to CNB somewhat justifies the conditional model, although the plots cannot be directly compared since the models use different number of features.

5 Conclusion

Ability to naturally handle sparsity is fast emerging as a key requirement for large scale data mining. In this paper, we presented a family of LD-CNB models that can not only handle sparsity, but actually take full advantage of it. On one hand, such models naturally extend the popular naive-Bayes models to work with sparse observations by conditioning the model on the observed features. On the other hand, we show that the machinery of hierarchical Bayesian modeling can be readily applied to such conditional naive-Bayes models. The stability of the performance of the LD-CNB model on held out test data is reassuring, and highlights the promise of the proposed family of models.

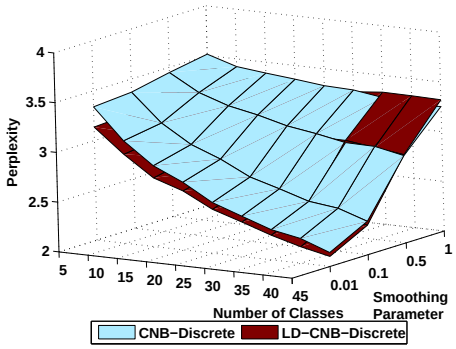


Figure 2. Perplexity of CNB-Discrete and LD-CNB-Discrete on training set over a range of number of classes and smoothing.

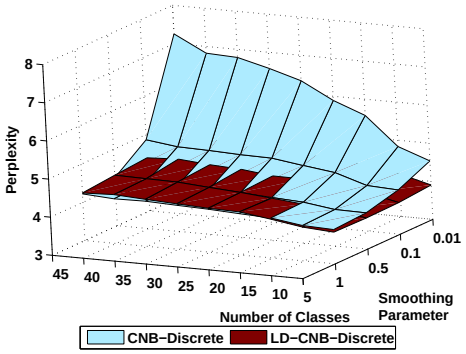


Figure 3. Perplexity of CNB-Discrete and LD-CNB-Discrete on testing set over a range of number of classes and smoothing.

Acknowledgements: The research was supported by DTC Data Mining Consortium and Grant-in-Aid at the University of Minnesota, Twin Cities.

References

[1] A. Banerjee, C. Krumpelman, S. Basu, R. Mooney, and J. Ghosh. Model-based overlapping clustering. In *KDD*, pages 532–537, 2005.

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

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

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

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

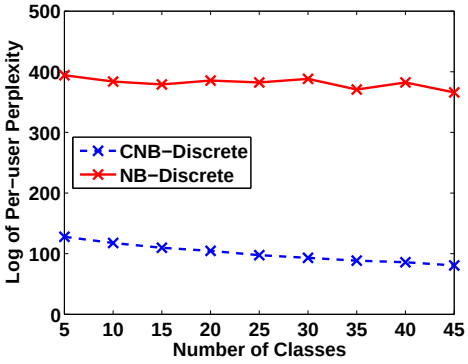


Figure 4. Log of peruser-perplexity of NB-Discrete and CNB-Discrete on training set with different number of classes while $\epsilon = 0.01$

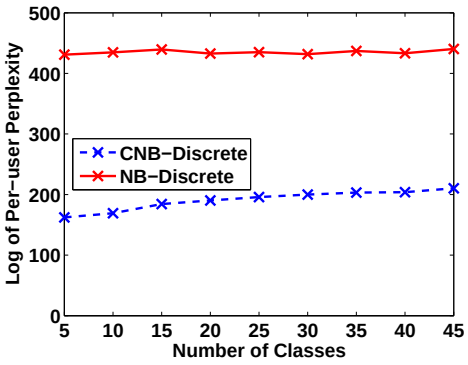


Figure 5. Log of peruser-perplexity of NB-Discrete and CNB-Discrete on testing set with different number of classes while $\epsilon = 0.01$

[6] T. Hoffman. Probabilistic latent semantic indexing. In *UAI*, 1999.

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

[8] Movielens. http://movielens.umn.edu.

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

[10] A. Ng and M. Jordan. On discrminative vs generative classifiers: A comparison of logistic regression and naive Bayes. In *NIPS*, 2001.

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

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

[13] E. Saund. Unsupervised learning of mixtures of multiple causes in binary data. In *NIPS*, 1994.

[14] M. Shahami, M. A. Hearst, and E. Saund. Applying the multiple cause model to text categorization. In *ICML*, 1996.