End-to-end Learning of Latent Dirichlet Allocation by Mirror-Descent Back Propagation

Jianshu Chen*, Ji He†, Yelong Shen*, Lin Xiao*, Xiaodong He*, Jianfeng Gao*, Xinying Song* and Li Deng*

*Microsoft Research, Redmond, WA 98052, USA,
{jianshuc,yeshen,lin.xiao,xiaohe,jfgao,xinson,deng}@microsoft.com
†Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA,
jvking@uw.edu

Abstract

We develop a fully discriminative learning approach for supervised Latent Dirichlet Allocation (LDA) model, which maximizes the posterior probability of the prediction variable given the input document. Different from traditional variational learning or Gibbs sampling approaches, the proposed learning method applies (i) the mirror descent algorithm for exact maximum a posterior inference and (ii) back propagation with stochastic gradient descent for model parameter estimation, leading to scalable learning of the model in an end-to-end discriminative manner. As a byproduct, we also apply this technique to develop a new learning method for the traditional unsupervised LDA model. Experimental results on two real-world regression and classification tasks show that the proposed methods significantly outperform the previous supervised/unsupervised LDA learning methods.

1 Introduction

Latent Dirichlet Allocation (LDA) [4], among various forms of topic models, is an important probabilistic generative model for analyzing large collections of text corpora. In LDA, each document is modeled as a collection of words, where each word is assumed to be generated from a certain topic drawn from a topic distribution. The topic distribution can be viewed as a latent representation of the document, which can be used as a feature for prediction purpose (e.g., sentiment analysis). In particular, the inferred topic distribution is fed into a separate classifier or regression model (e.g., logistic regression or linear regression) to perform prediction. Such a separate learning structure usually significantly restricts the performance of the algorithm. For this purpose, various supervised topic models have been proposed to model the documents jointly with the label information. In [3], variational methods was applied to learn a supervised LDA (sLDA) model by maximizing the lower bound of the joint probability of the input data and the labels. The DiscLDA method developed in [11] learns the transformation matrix from the latent topic representation to the output in a discriminative manner, while learning the topic to word distribution in a generative manner similar to the standard LDA. In [21, 22], max margin supervised topic models are developed for classification and regression, which are trained by optimizing the sum of the variational bound for the log marginal likelihood and an additional term that characterizes the prediction margin. These methods successfully incorporate the information from both the input data and the labels, and showed better performance in prediction compared to the vanilla LDA model.

One challenge in LDA is that the exact inference is intractable, i.e., the posterior distribution of the topics given the input document cannot be evaluated explicitly. For this reason, various approximate inference methods are proposed, such as variational learning [3,4,21,22] and Gibbs sampling [7,23], for computing the approximate posterior distribution of the topics. In this paper, we will show that,

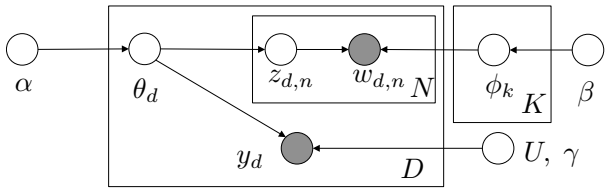


Figure 1: Graphical representation of the supervised LDA model. Shaded nodes are observables.

although the full posterior probability of the topic distribution is difficult, its maximum a posteriori (MAP) inference, as a simplified problem, is a convex optimization problem when the Dirichlet parameter satisfies certain conditions, which can be solved efficiently by the mirror descent algorithm (MDA) [1,15,18]. Indeed, Sontag and Roy [16] pointed out that the MAP inference problem of LDA in this situation is polynomial-time and can be solved by an exponentiated gradient method, which shares a same form as our mirror-descent algorithm with constant step-size. Nevertheless, different from [16], which studied the inference problem alone, our focus in this paper is to integrate back propagation with mirror-descent algorithm to perform fully discriminative training of supervised topic models, as we proceed to explain below.

Among the aforementioned methods, one training objective of the supervised LDA model is to maximize the joint likelihood of the input and the output variables [3]. Another variant is to maximize the sum of the log likelihood (or its variable bound) and a prediction margin [21–23]. Moreover, the DiscLDA optimizes part of the model parameters by maximizing the marginal likelihood of the input variables, and optimizes the other part of the model parameters by maximizing the conditional likelihood. For this reason, DiscLDA is not a fully discriminative training of all the model parameters. In this paper, we propose a fully discriminative training of all the model parameters by maximizing the posterior probability of the output given the input document. We will show that the discriminative training can be performed in a principled manner by naturally integrating the back-propagation with the MDA-based exact MAP inference. Discriminative training of generative model is widely used and usually outperforms standard generative training in prediction tasks [2, 6, 9, 10, 12, 20]. As pointed out in [2, 12], discriminative training increases the robustness against the mismatch between the generative model and the real data. To our best knowledge, this paper is the first work to perform a fully end-to-end discriminative training of LDA. Experimental results on two real-world tasks also show the superior performance of discriminative training of LDA models on text analysis.

In addition to the aforementioned related studies on unsupervised and supervised LDA models [3,11, 21–23], there have been another stream of work that applied empirical risk minimization to graphical models such as Markov Random Field (MRF) and nonnegative matrix factorization (NMF) [8, 17]. Specifically, in [17], an approximate inference algorithm, belief propagation, is used to compute the belief of the output variables, which is further fed into a decoder to produce the prediction. The approximate inference and the decoder are treated as an entire black-box decision rule, which is tuned jointly via back propagation. Our work is different from the above studies in that we use an exact MAP inference based on convex optimization theory motivate the discriminative training from a principled probabilistic framework.

2 Smoothed Supervised LDA Model

We consider the smoothed supervised LDA model in Figure 1. Let K be the number of topics, N be the number of words in each document, V be the vocabulary size, and D be the number of documents in the corpus. The generative process of the model in Figure 1 can be described as:

1. For each document d, choose the topic proportions according to a Dirichlet distribution: $\theta_d \sim p(\theta_d|\alpha) = \mathrm{Dir}(\alpha)$, where α is a $K \times 1$ vector consisting of nonnegative components.
2. Draw each column ϕ_k of a $V \times K$ matrix Φ independently from an exchangeable Dirichlet distribution: $\phi_k \sim \mathrm{Dir}(\beta)$ (i.e., $\Phi \sim p(\Phi|\beta)$), where $\beta > 0$ is the smoothing parameter.
3. To generate each word $w_{d,n}$:

(a) Choose a topic $z_{d,n} \sim p(z_{d,n}|\theta_d) = \text{Multinomial}(\theta_d)$. [1]
(b) Choose a word $w_{d,n} \sim p(w_{d,n}|z_{d,n}, \Phi) = \text{Multinomial}(\phi_{z_{d,n}})$.

4. Choose the $C \times 1$ response vector: $y_d \sim p(y_d|\theta, U, \gamma)$.
   (a) In regression, $p(y_d|\theta_d, U, \gamma) = N(U\theta_d, \gamma^{-1})$, where U is a $C \times K$ matrix consisting of regression coefficients.
   (b) In multi-class classification, $p(y_d|\theta_d, U, \gamma) = \text{Multinomial}\big(\sigma(\gamma U\theta_d)\big)$, where $\sigma : \mathbb{R}^C \to \mathbb{R}^C$ is a softmax function defined as $\sigma(x)_c = \frac{e^{x_c}}{\sum_{c'=1}^{C} e^{x_{c'}}}$, $c = 1, \ldots, C$.

Therefore, the entire model can be described by the following joint probability

$$p(\Phi|\beta) \prod_{d=1}^{D} \Big[\underbrace{p(y_d|\theta_d, U, \gamma) \cdot p(\theta_d|\alpha) \cdot p(w_{d,1:N}|z_{d,1:N}, \Phi) \cdot p(z_{d,1:N}|\theta_d)}_{\triangleq p(y_d, \theta_d, w_{d,1:N}, z_{d,1:N}|\Phi, U, \alpha, \gamma)} \Big] \tag{1}$$

where $w_{d,1:N}$ and $z_{d,1:N}$ denotes all the words and the associated topics, respectively, in the d-th document. Note that the model in Figure 1 is slightly different from the one proposed in [3], where, in addition to the Dirichlet smoothing part on ϕ_k, the response variable y_d in Figure 1 is coupled with θ_d instead of $z_{d,1:N}$ as in [3]. Blei and Mcauliffe also pointed out this choice as an alternative in [3]. We will show that this modification will enable us to develop an end-to-end discriminative training with superior prediction performance.

To develop a fully discriminative training method for the model parameters Φ and U, we follow the argument in [2, 12], which states that the discriminative training is also equivalent to maximizing the joint likelihood of a new model family with an additional set of parameters:

$$\arg\max_{\Phi, U, \tilde{\Phi}} p(\Phi|\beta) p(\tilde{\Phi}|\beta) \prod_{d=1}^{D} p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma) \prod_{d=1}^{D} p(w_{d,1:N}|\tilde{\Phi}, \alpha) \tag{2}$$

where $p(w_{d,1:N}|\tilde{\Phi}, \alpha)$ is obtained by marginalizing $p(y_d, \theta_d, w_{d,1:N}, z_{d,1:N}|\Phi, U, \alpha, \gamma)$ in (1) and replace Φ with $\tilde{\Phi}$. The above problem (2) decouples into

$$\arg\max_{\Phi, U} \Big[\ln p(\Phi|\beta) + \sum_{d=1}^{D} \ln p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma) \Big] \tag{3}$$

$$\arg\max_{\tilde{\Phi}} \Big[\ln p(\tilde{\Phi}|\beta) + \sum_{d=1}^{D} \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha) \Big] \tag{4}$$

which are the discriminative learning problem of supervised LDA (Eq. (3)), and the unsupervised learning problem of LDA (Eq. (4)), respectively. It was pointed out in [2] that the discriminative training (3) improves performance by compensating for model mismatch, i.e., the differences between the true distribution of the data and the distribution specified by the model. We will show that both problems can be solved in a unified manner using a new MAP inference and back propagation.

3 Exact MAP Inference

We first consider the inference problem in the smoothed LDA model. For the supervised case, the main objective is to infer y_d given the words $w_{d,1:N}$ in each document d, i.e., computing

$$p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma) = \int_{\theta_d} p(y_d|\theta_d, U, \gamma) p(\theta_d|w_{d,1:N}, \Phi, \alpha) d\theta_d \tag{5}$$

where the probability $p(y_d|\theta_d, U, \gamma)$ is known (e.g., multinomial or Gaussian for classification and regression problems — see Section 2). The main challenge is to evaluate $p(\theta_d|w_{d,1:N}, \Phi, \alpha)$, i.e., infer the topic proportion given each document, which is also the important inference problem in

[1] We will represent all the multinomial variables by a one-hot vector that has a single component equal to one and all other components being zero, where the position of the one is determined by the value of the multinomial variable.

the unsupervised LDA model. However, it is well known that the exact evaluation of the posterior probability $p(\theta_d|w_{d,1:N}, \Phi, \alpha)$ is intractable [3,4,7,11,21–23]. For this reason, various approximate inference methods, such as variational inference [3, 4, 11, 21, 22] and Gibbs sampling [7, 23], have been proposed to compute the approximate posterior probability. In this paper, we take an alternative approach for inference; given each document d, we only seek a point (maximum a posterior) estimate of θ_d, instead of its full (approximate) posterior probability. The major motivation is that, although the full posterior probability of θ_d is difficult, its MAP inference, as a simplified problem, is a convex optimization problem (Section 3.1) and is thus tractable. Furthermore, having the MAP estimate of θ_d, we can efficiently infer the prediction variable y_d according to the following approximation of $p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma)$ from (5):

$$p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma) = \mathbb{E}_{\theta_d|w_{d,1:N}}\left[p(y_d|\theta_d, U, \gamma)\right] \approx p(y_d|\hat{\theta}_{d|w_{d,1:N}}, U, \gamma) \tag{6}$$

where $\mathbb{E}_{\theta_d|w_{d,1:N}}[\cdot]$ denotes the conditional expectation with respect to θ_d given $w_{d,1:N}$, and the expectation is sampled by the MAP estimate, $\hat{\theta}_{d|w_{d,1:N}}$, of θ_d given $w_{d,1:N}$, defined as

$$\hat{\theta}_{d|w_{d,1:N}} = \arg\max_{\theta_d} p(\theta_d|w_{d,1:N}, \Phi, \alpha, \beta) \tag{7}$$

The approximation gets more precise when $p(\theta_d|w_{d,1:N}, \Phi, \alpha, \beta)$ becomes more concentrated around $\hat{\theta}_{d|w_{d,1;N}}$. Experimental results on several real datasets (Section 5) show that the approximation (6) provides excellent prediction performance.

3.1 MAP Inference as a Convex Optimization Problem

Using the Bayesian rule $p(\theta_d|w_{d,1:N}, \Phi, \alpha) = p(\theta_d|\alpha)p(w_{d,1:N}|\theta_d, \Phi)/p(w_{d,1:N}|\Phi, \alpha)$ and the fact that $p(w_{d,1:N}|\Phi, \alpha)$ is independent of θ_d, we obtain the equivalent form of (7) as

$$\hat{\theta}_{d|w_{d,1:N}} = \arg\max_{\theta_d \in \mathcal{P}_K} \left[\ln p(\theta_d|\alpha) + \ln p(w_{d,1:N}|\theta_d, \Phi)\right] \tag{8}$$

where $\mathcal{P}_K = \{\theta \in \mathbb{R}^K : \theta_j \geq 0, \sum_{j=1}^K \theta_j = 1\}$ denotes the $(K-1)$-dimensional probability simplex vector, $p(\theta_d|\alpha)$ is the Dirichlet distribution described earlier, and $p(w_{d,1:N}|\theta_d, \Phi)$ can be computed by integrating $p(w_{d,1:N}, z_{d,1:N}|\theta_d, \Phi) = \prod_{n=1}^N p(w_{d,n}|z_{d,n}, \Phi)p(z_{d,n}|\theta_d)$ over $z_{d,1:N}$, which leads to (derived in Appendix A)

$$p(w_{d,1:N}|\theta_d, \Phi) = \prod_{v=1}^V \left(\sum_{j=1}^K \theta_{d,j}\Phi_{vj}\right)^{x_{d,v}} = p(x_d|\theta_d, \Phi) \tag{9}$$

where $x_{d,v}$ denotes the term frequency of the v-th word (in vocabulary) inside the d-th document, and x_d denotes the V-dimensional bag-of-words (BoW) vector of the d-th document. Note that $p(w_{d,1:N}|\theta_d, \Phi)$ depends on $w_{d,1:N}$ only via the BoW vector x_d, which is the sufficient statistics. Therefore, we use $p(x_d|\theta_d, \Phi)$ and $p(w_{d,1:N}|\theta_d, \Phi)$ interchangeably from now on. Substituting the expression of Dirichlet distribution and (9) into (8), we get

$$\begin{aligned}\hat{\theta}_{d|w_{d,1:N}} &= \arg\max_{\theta_d \in \mathcal{P}_K} \left[x_d^T \ln(\Phi\theta_d) + (\alpha - \mathbb{1})^T \ln\theta_d\right] \\ &= \arg\min_{\theta_d \in \mathcal{P}_K} \left[-x_d^T \ln(\Phi\theta_d) - (\alpha - \mathbb{1})^T \ln\theta_d\right]\end{aligned} \tag{10}$$

where we dropped the terms independent of θ_d, and $\mathbb{1}$ denotes an all-one vector. Note that when each element of α is greater than or equal to one, the objective function in (10) is convex and the problem is convex. When α is strictly greater than one, the objective function is strictly convex and has a unique solution. In this paper, we will only focus on the regime of α being greater than one.

3.2 Mirror Descent Algorithm for MAP Inference

An efficient approach to solving the constrained optimization problem (10) is the mirror descent algorithm (MDA) with Bregman divergence chosen to be generalized Kullback-Leibler divergence [1, 15, 18]. Specifically, let $f(\theta_d)$ denote the cost function in (10), then the MDA updates the MAP estimate of θ_d iteratively according to:

$$\theta_{d,\ell} = \arg\min_{\theta_d \in \mathcal{P}_K} \left[f(\theta_{d,\ell-1}) + [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T(\theta_d - \theta_{d,\ell-1}) + \frac{1}{T_{d,\ell}}\Psi(\theta_d, \theta_{d,\ell-1})\right] \tag{11}$$

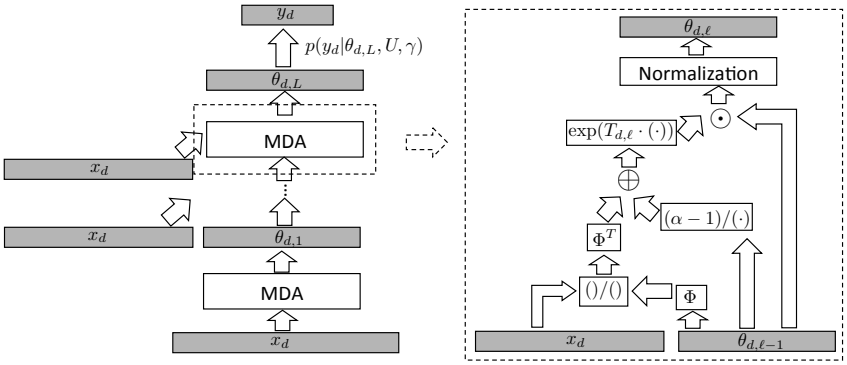


Figure 2: Layered architecture for computing $p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma)$, where $()/()$ denotes element-wise division, $\odot$ denotes Hadamard product, and $\exp()$ denotes element-wise exponential.

$\theta_{d,\ell}$ denotes the estimate of $\theta_{d,\ell}$ at the ℓ-th iteration, $T_{d,\ell}$ denotes the step-size of MDA, and $\Psi(x, y)$ is the Bregman divergence chosen to be $\Psi(x, y) = x^T \ln(x/y) - \mathbb{1}^T x + \mathbb{1}^T y$. The argmin in (11) can be solved in closed-form (see Appendix B) as

$$\theta_{d,\ell} = \frac{1}{C_\theta} \cdot \theta_{d,\ell-1} \odot \exp\left(T_{d,\ell}\left[\Phi^T \frac{x_d}{\Phi\theta_{d,\ell-1}} + \frac{\alpha - \mathbb{1}}{\theta_{d,\ell-1}}\right]\right),\ \ell = 1, \ldots, L, \quad \theta_{d,0} = \frac{1}{K}\mathbb{1} \tag{12}$$

where C_θ is a normalization factor such that $\theta_{d,\ell}$ adds up to one, $\odot$ denotes Hadamard product, L is the number of MDA iterations, and the divisions in (12) are element-wise operations. Note that the recursion (12) naturally enforces each $\theta_{d,\ell}$ to be on the probability simplex. The MDA step-size $T_{d,\ell}$ can be either constant, i.e., $T_{d,\ell} = T$, or adaptive over iterations and samples, determined by line search (see Appendix C). The computation complexity in (12) is low since most computations are sparse matrix operations. For example, although by itself $\Phi\theta_{d,\ell-1}$ in (12) is a dense matrix multiplication, we only need to evaluate the elements of $\Phi\theta_{d,\ell-1}$ at the positions where the corresponding elements of x_d are nonzero, because all other elements of $x_d/\Phi\theta_{d,\ell-1}$ is known to be zero. Overall, the computation complexity in each iteration of (12) is $O(\mathrm{nTok} \cdot K)$, where nTok denotes the number of unique tokens in the document. In practice, we only use a small number of iterations, L, in (12) and use $\theta_{d,L}$ to approximate $\hat{\theta}_{d|w_{d,1:N}}$ so that (6) becomes

$$p(y_d|w_{d,1:N}, \Phi, U, \alpha, \gamma) \approx p(y_d|\theta_{d,L}, U, \gamma) \tag{13}$$

In summary, the inference of θ_d and y_d can be implemented by the layered architecture in Figure 2, where the top layer infers y_d using (13) and the MDA layers infer θ_d iteratively using (12). Figure 2 also implies that the the MDA layers act as a feature extractor by generating the MAP estimate $\theta_{d,L}$ for the output layer. Our end-to-end learning strategy developed in the next section jointly learns the model parameter U at the output layer and the model parameter Φ at the feature extractor layers to maximize the posterior of the prediction variable given the input document.

4 Learning by Mirror-Descent Back Propagation

We now consider the supervised learning problem (3) and the unsupervised learning problem (4), respectively, using the developed MDA-based MAP inference. We first consider the supervised learning problem. With (13), the discriminative learning problem (3) can be approximated by

$$\arg\min_{\Phi, U}\left[-\ln p(\Phi|\beta) - \sum_{d=1}^{D} \ln p(y_d|\theta_{d,L}, U, \gamma)\right] \tag{14}$$

which can be solved by stochastic gradient descent (SGD). Note that the cost function in (14) depends on U explicitly through $p(y_d|\theta_{d,L}, U, \gamma)$, which can be computed directly from its definition in Sec. 2. On the other hand, the cost function in (14) depends on Φ implicitly through $\theta_{d,L}$. From Figure 2, we observe that $\theta_{d,L}$ not only depends on Φ explicitly (as indicated in the MDA block on the right-hand side of Figure 2) but also depends on Φ implicitly via $\theta_{d,L-1}$, which in turn depends on Φ both explicitly and implicitly (through $\theta_{d,L-2}$) and so on. That is, the dependency of

the cost function on Φ is in a layered manner. Therefore, we devise a back propagation procedure to efficiently compute its gradient with respect to Φ according to the mirror-descent graph in Figure 2, which back propagate the error signal through the MDA blocks at different layers. The gradient formula and the implementation details of the learning algorithm can be found in Appendices C–D.

For the unsupervised learning problem (4), the gradient of $\ln p(\tilde{\Phi}|\beta)$ with respect to $\tilde{\Phi}$ assumes the same form as that of $\ln p(\Phi|\beta)$. Moreover, it can be shown that the gradient of $\ln p(w_{d,1:N}|\tilde{\Phi}, \alpha, \gamma)$ with respect $\tilde{\Phi}$ can be expressed as (see Appendix E):

$$\frac{\partial \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha)}{\partial \tilde{\Phi}} = \mathbb{E}_{\theta_d|x_d}\left\{\frac{\partial}{\partial \tilde{\Phi}}\left[\ln p(x_d|\theta_d, \tilde{\Phi}) + \ln p(\theta_d|\alpha)\right]\right\} \tag{15}$$

where $p(x_d|\theta_d, \tilde{\Phi})$ assumes the same form as (9) except Φ is replaced by $\tilde{\Phi}$. The conditional expectation is evaluated with respect to the posterior probability $p(\theta_d|w_{d,1:N}, \tilde{\Phi}, \alpha)$, which can be sampled by the MAP estimate of θ_d:

$$\frac{\partial \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha)}{\partial \tilde{\Phi}} \approx \frac{\partial}{\partial \tilde{\Phi}}\left[\ln p(x_d|\theta_{d,L}, \tilde{\Phi}) + \ln p(\theta_{d,L}|\alpha)\right] \tag{16}$$

where $\theta_{d,L}$ is an approximation of $\hat{\theta}_{d|w_{d,1:N}}$ computed via (12) and Figure 2.

5 Experiments

5.1 Description of Datasets and Baselines

We evaluated our proposed supervised learning (denoted as BP-sLDA) and unsupervised learning (denoted as BP-LDA) methods on two real-world datasets. The first dataset we use is a large-scale dataset built on Amazon movie reviews (AMR) [13]. The data set consists of 7.9 million movie reviews (1.48 billion words) from Amazon, written by 889,176 users, on a total of 253,059 movies. For text preprocessing we removed punctuations and lowercasing capital letters. A vocabulary of size 5,000 is built by selecting the most frequent words. Same as [19], we shifted the review scores so that they have zero mean. The task is formulated as a regression problem, where we seek to predict the rating score using the text of the review.

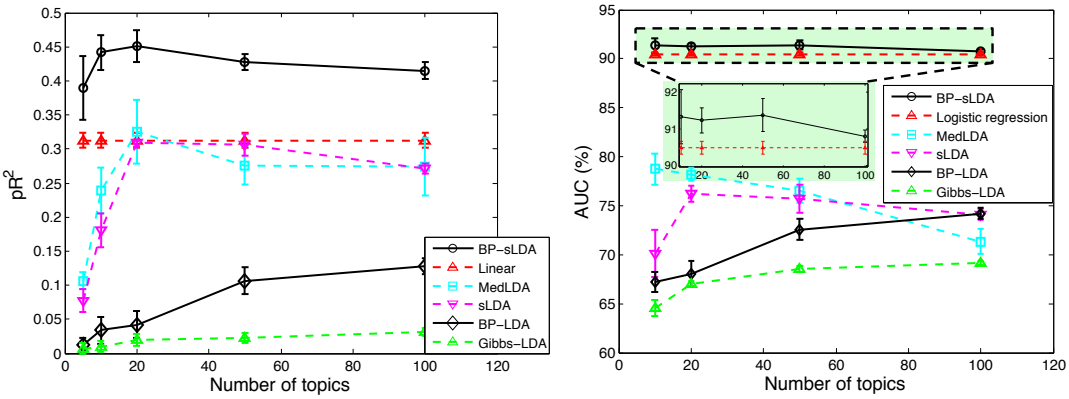
Second, we demonstrate the effectiveness of our algorithm on a multi-domain sentiment (MultiSent) classification task. Sentiment classification system has gained popularity due to their application to multiple text genres, including financial news and product reviews. We use the dataset provided by [5], which contains a total 342,104 product reviews consisting of 25 types of product reviews, such as apparel, electronics, kitchen and housewares. The task is formulated as a binary classification problem to predict the polarity (positive or negative) of each review. Similar to AMR task, we preprocessed the text by removing punctuations and lowercasing capital letters. A vocabulary of size 1,000 is built from the most frequent words.

We examined our proposed methods (BP-sLDA and BP-LDA) as well as baselines on both tasks. For BP-sLDA, $p(y_d|\theta_d, U, \gamma)$ is chosen to be Gaussion on the AMR regression task and multinormial on the MultiSent classification task (see Sec. 2). For BP-LDA, we first train the models in an unsupervised manner, and then generate per-document topic proportion θ_d as their features in the inference steps, on top of which we train a linear regression model in AMR regression task and train a logistic regression model in the MultiSent classification task, respectively. The baseline algorithms are implemented either in C++ or Java and our proposed algorithms are implemented in C#.[2] We compared our methods to the unsupervised LDA learned by Gibbs sampling (Gibbs-LDA) [14], logistic/linear regression using raw bag-of-words (BoW), supervised-LDA (sLDA) [3], and MedLDA [21, 22]. Similar to BP-LDA, a separate linear/logistic regression is trained on the features generated by Gibbs-LDA. All the experiments are conducted with 5-fold cross validation.

5.2 Prediction Performance

We first evaluate the prediction performance of different models on the AMR regression task. We use the predictive R^2 to measure the prediction performance, defined as: $pR^2 = 1 - (\sum_d (y_d^o -$

[2]The code will be released soon.



(a) pR^2 scores on AMR regression task (79K) (b) AUC on MultiSent classification task

Figure 3: Prediction performance on AMR regression task (measured in pR^2) and MultiSent classification task (measured in AUC). Higher score is better for both, with perfect value being one.

Table 1: pR^2 on full AMR data (7.9M documents and 5K vocabulary size).

Number of topics	5	10	20	50	100
Linear regression	0.384 ± 0.001				
BP-sLDA ($\alpha = 1.001$)	0.528±0.021	0.562±0.016	0.577±0.009	0.613±0.017	0.641±0.028

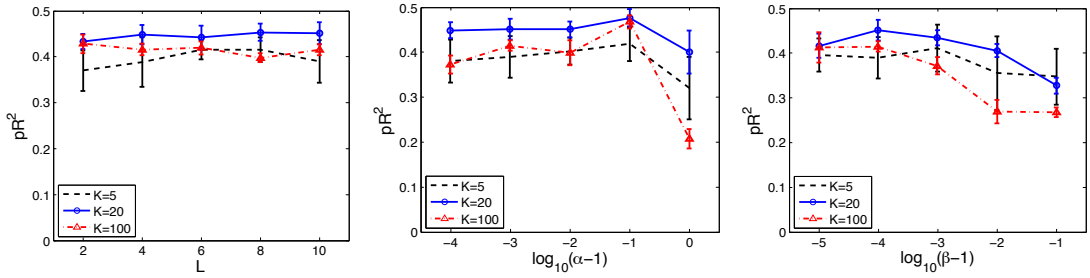
$y_d)^2)/(\sum_d (y_d^o - \bar{y}_d^o)^2)$, where y_d^o denotes the label of the d-th document in the heldout (out-of-fold) set during the 5-fold cross validation, $\bar{y}_d^o$ is the mean of all y_d^o in the heldout set, and y_d is the predicted value. We first created a subset by randomly sampling 79K documents (reviews) from the 7.9 million reviews. The pR^2 scores of different models with varying number of topics are shown in Figure 3(a). Note that the BP-sLDA model outperforms the other baselines with large margin. Moreover, the unsupervised BP-LDA model outperforms the unsupervised LDA model trained by Gibbs sampling (Gibbs-LDA). We further train our BP-sLDA model on the full 7.9M dataset with 5-fold cross validation and list the pR^2 scores in Table 1. We can see that pR^2 improves significantly compared to the best results on the 79K dataset shown in Figure 3(a). Moreover, the results in Table 1 also significantly outperform the pR^2 scores of Gibbs-sLDA [23], Spectral-sLDA [19], and the Hybrid method (Gibbs-sLDA initialized with Spectral-sLDA) reported in [19], whose pR^2 scores are between 0.1 and 0.2 for $5 \sim 10$ topics (and deteriorate when further increasing the topic number). The results therein are obtained on the same full AMR data with same setting as this paper. To further demonstrate the superior performance of BP-sLDA on the large vocabulary scenario, we trained BP-sLDA on the full 7.9M AMR dataset with full vocabulary (701K) and obtain the pR^2 scores in Table 2. Note that the results are even significantly better than our results in Table 1.

Next, we evaluate the performance of our algorithms on the binary classification task of multi-domain sentiment analysis. We use the area-under-the-curve (AUC) of the operating curve of probability of correct positive versus probability of false positive as our performance metric. In Figure 3(b), we show the AUC of our methods and the baselines, which also shows that BP-sLDA outperforms other methods and that BP-LDA outperforms the unsupervised Gibbs-LDA model.

From Figure 3, we note that the BP-sLDA model also consistently outperforms the linear regression or logistic regression model on the raw bag-of-words features. In Fig.3b (MultiSent), logistic regression achieves AUC of 90.4%, while BP-sLDA achieves the best AUC of 91.4% with 20 topics, which is about 10% relative improvement over logistic regression. And BP-sLDA significantly outperforms prior-art topic models, which have AUCs less than 80%. This means that our proposed discriminative training method and MDA-based MAP inference together are able to extract useful features from the raw BoW inputs for prediction purpose.

Table 2: pR^2 on full AMR data (7.9M documents and 701K vocabulary size).

Number of topics	5	10	20	50	100
Linear regression	0.403				
BP-sLDA	0.633	0.677	0.672	0.682	**0.684**



(a) Number of MDA iterations L (b) Dirichlet parameter α (c) Smoothing parameter β

Figure 4: Sensitivity of hyper parameters: pR^2 score for different L, α, and β.

In addition, we also conducted a new binary text classification experiment with highly promising results on a large-scale proprietary dataset for business-centric applications (1.2M documents and vocabulary size of 128K). In this new task, BP-sLDA (200 topics) achieves AUC of 92.2% and error rate of 15.2%, while LR has AUC of 90.5% and error rate of 17.1% (11% relative error rate cut). The gain is consistent with what was observed in the other two tasks.

5.3 Analysis and Discussion

We now analyze the influence of different hyper parameters on the prediction performance. Note from Figure 3(a) that, when we increase the number of topics, the pR^2 score of BP-sLDA first improves and then slightly deteriorates after it goes beyond 20 topics. This is most likely to be caused by overfitting on the small dataset (79K documents), because the BP-sLDA models trained on the full 7.9M dataset produce much higher pR^2 scores (Table 1) than that on the 79K dataset and keep improving as the model size (number of topics) increases. Another interesting observation from Figure 3 is that, with limited amount of labeled data, the unsupervised LDA models (BP-LDA and Gibbs-LDA) are less prone to overfitting. Since unlabeled data are widely available, one future work is to combine the supervised and unsupervised parts together to have a semi-supervised LDA models. The framework suggested by [2, 12] could be one potential approach to integrate these two parts together.

To further understand the influence of the other hyper-parameters, we plot in Figure 4 the pR^2 scores of BP-sLDA on the 79K AMR dataset for different values of L, α, and β. The performance is not very sensitive to the number of MDA inference steps L. One explanation for this phenomena is that the mirror-descent back propagation, as an end-to-end training of the prediction output, compensates the imperfection caused by the limited number of inference steps. And, we observe that, by properly tuning the Dirichlet parameter α and the smoothing parameter β, we could further improve the prediction performance of the model. Moreover, although we mainly focus on convex inference under $\alpha > 1$, our algorithm could also handle $\alpha < 1$ case except that, in this case, the inference is no longer convex and hence no global optimal MAP inference is guaranteed as for the methods prior to this work. Table 3 shows the corresponding pR^2 scores of BP-sLDA on the 7.9M AMR dataset. Although the results is not as good as the $\alpha > 1$ case in Table 1, they still significantly outperform the baselines.

5.4 Efficiency in Computation Time

To compare the efficiency of the algorithms, we show the training time (in hours) of different models on the AMR dataset (79K and 7.9M) in Figure 5, which shows that our algorithm scales well when

Table 3: pR^2 for $\alpha < 1$ case on full AMR data (7.9M documents and 5K vocabulary size).

Number of topics	5	10	20	50	100
BP-sLDA ($\alpha = 0.5$)	0.488	0.548	0.575	0.571	0.574
BP-sLDA ($\alpha = 0.1$)	0.441	0.558	0.572	0.569	0.570

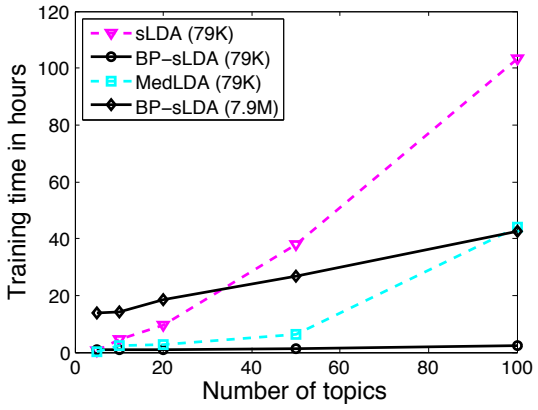


Figure 5: Training time of different methods (in hours) on the AMR dataset.

we increase the model size (number of topics). In addition, it also scales well on the large-scale (7.9M) dataset, which can be completed within reasonable amount of time.

6 Conclusion

We have developed novel learning approaches for both supervised and unsupervised LDA models, using exact MAP inference with mirror descent algorithm and back propagation. In particular, the supervised LDA model is trained in an end-to-end fully discriminative manner by maximizing the posterior probability of the prediction variable given input documents. We evaluate the prediction performance of the models on two real-world regression and classification tasks. The results show that the discriminative training approach significantly improves the performance of the supervised LDA model relative to previous learning methods. Moreover, the newly developed inference and learning techniques also improve the performance of the unsupervised LDA model by providing better features for prediction. Future works include (i) exploring other optimization algorithms for the MAP inference problem, such as accelerated mirror descent, and (ii) developing semi-supervised learning of LDA based on the framework suggested by [2, 12]. More importantly, note that the layered architecture in Figure 2 could also be viewed as a deep feedforward neural network with special structures designed from the topic model in Figure 1. This opens up a new direction of combining the strength of both (generative) topic models and neural networks to develop new deep learning models that are scalable, interpretable and having high prediction performance.

References

[1] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[2] C. M. Bishop and J. Lasserre. Generative or discriminative? getting the best of both worlds. *Bayesian Statistics*, 8:3–24, 2007.

[3] D. M. Blei and J. D. Mcauliffe. Supervised topic models. In *Proc. NIPS*, pages 121–128, 2007.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. ACL*, volume 7, pages 440–447, 2007.

[6] G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In *Proc. COMPSTAT*, pages 721–728, 2004.

[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc. of the National Academy of Sciences*, pages 5228–5235, 2004.

[8] J. R. Hershey, J. L. Roux, and F. Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv:1409.2574*, 2014.

[9] A. Holub and P. Perona. A discriminative framework for modelling object classes. In *Proc. IEEE CVPR*, volume 1, pages 664–671, 2005.

[10] S. Kapadia. *Discriminative Training of Hidden Markov Models*. PhD thesis, University of Cambridge, 1998.

[11] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Proc. NIPS*, pages 897–904, 2008.

[12] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *Proc. IEEE CVPR*, volume 1, pages 87–94, 2006.

[13] J. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proc. WWW*, pages 897–908, 2013.

[14] Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.

[15] D. B. Nemirovsky. A. S., Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.

[16] D. Sontag and D. Roy. Complexity of inference in latent dirichlet allocation. In *Proc. NIPS*, pages 1008–1016, 2011.

[17] V. Stoyanov, A. Ropson, and J. Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *Proc. AISTATS*, pages 725–733, 2011.

[18] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Optimization*, 2008.

[19] Y. Wang and J. Zhu. Spectral methods for supervised topic models. In *Proc. NIPS*, pages 1511–1519, 2014.

[20] Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. Discriminatively trained Markov model for sequence classification. In *Proc. IEEE ICDM*, 2005.

[21] J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proc. ICML*, pages 1257–1264, 2009.

[22] J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models. *JMLR*, 13(1):2237–2278, 2012.

[23] J. Zhu, N. Chen, H. Perkins, and B. Zhang. Gibbs max-margin topic models with data augmentation. *JMLR*, 15(1):1073–1110, 2014.

Appendices

A Derivation of $p(w_{d,1:N}|\theta_d, \Phi)$

To derive $p(w_{d,1:N}|\theta_d, \Phi)$, we first write $p(w_{d,1:N}, z_{d,1:N}|\theta_d, \Phi)$ as

$$p(w_{d,1:N}, z_{d,1:N}|\theta_d, \Phi) = \prod_{n=1}^{N} p(w_{d,n}|z_{d,n}, \Phi)p(z_{d,n}|\theta_d) \tag{17}$$

The expression $p(w_t|\Phi, \theta_t)$ can be evaluated in closed-form by marginalizing out $\{z_{d,n}\}_{n=1}^{N}$ in the above expression:

$$\begin{aligned}
p(w_{d,1:N}|\theta_d, \Phi) &= \sum_{z_{d,1}} \cdots \sum_{z_{d,N}} \prod_{n=1}^{N} p(z_{d,n}|\theta_d) \cdot p(w_{d,n}|z_{d,n}, \Phi) \\
&= \prod_{n=1}^{N} \sum_{z_{d,n}} p(z_{d,n}|\theta_d) \cdot p(w_{d,n}|z_{d,n}, \Phi) \\
&= \prod_{n=1}^{N} \sum_{z_{d,n}} \left(\prod_{j=1}^{K} \theta_{d,j}^{z_{d,n,j}} \right) \left(\prod_{v=1}^{V} \prod_{j=1}^{K} \Phi_{vj}^{z_{d,n,j}\ w_{d,i,v}} \right) \\
&= \prod_{n=1}^{N} \sum_{z_{d,n}} \left(\prod_{v=1}^{V} \prod_{j=1}^{K} \theta_{d,j}^{z_{d,n,j}} \Phi_{vj}^{z_{d,n,j}\ w_{d,n,v}} \right) \\
&= \prod_{n=1}^{N} \left(\sum_{j=1}^{K} \theta_{d,j} \Phi_{vj} \right)^{w_{d,n,v}} \\
&= \prod_{v=1}^{V} \left(\sum_{j=1}^{K} \theta_{d,j} \Phi_{vj} \right)^{x_{d,v}}
\end{aligned} \tag{18}$$

where $w_{d,n,v}$ denotes the v-th element of the $V \times 1$ one-hot vector $w_{d,n}$, $w_{d,n}$ denotes the n-th word (token) inside the d-th document, and $x_{d,v}$ denotes the term frequency of the v-th word (in the vocabulary) inside the d-th document.

B Derivation of the Recursion for Mirror Descent Algorithm

First, we rewrite the optimization problem (11) as

$$\min_{\theta_d} \quad [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T(\theta_d - \theta_{d,\ell-1}) + \frac{1}{T_{d,\ell}} \Psi(\theta_d, \theta_{d,\ell-1}) \tag{19}$$

$$\text{s.t.} \quad \mathbb{1}^T\theta_d = 1, \quad \theta_d \succeq 0 \tag{20}$$

where $\theta_d \succeq 0$ denotes that each element of the vector θ_d is greater than or equal to zero. Using the fact that $\Psi(x, y) = x^T \ln(x/y) - \mathbb{1}^T x + \mathbb{1}^T y$, the constrained optimization problem (19)–(20) becomes

$$\min_{\theta_d} \quad [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T(\theta_d - \theta_{d,\ell-1}) + \frac{1}{T_{d,\ell}} \left[\theta_d^T \ln \frac{\theta_d}{\theta_{d,\ell-1}} - \mathbb{1}^T\theta_d + \mathbb{1}^T\theta_{d,\ell-1} \right] \tag{21}$$

$$\text{s.t.} \quad \mathbb{1}^T\theta_d = 1, \quad \theta_d \succeq 0 \tag{22}$$

Dropping the terms independent of θ_d, we can write (21)–(22) as

$$\min_{\theta_d} \quad [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T\theta_d + \frac{1}{T_{d,\ell}} \left[\theta_d^T \ln \frac{\theta_d}{\theta_{d,\ell-1}} - \mathbb{1}^T\theta_d \right] \tag{23}$$

$$\text{s.t.} \quad \mathbb{1}^T\theta_d = 1, \quad \theta_d \succeq 0 \tag{24}$$

To solve (23)–(24), we write its Lagrangian as

$$L = [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T \theta_d + \frac{1}{T_{d,\ell}} \left[\theta_d^T \ln \frac{\theta_d}{\theta_{d,\ell-1}} - \mathbb{1}^T \theta_d \right] + \lambda(\mathbb{1}^T \theta_d - 1) \tag{25}$$

where we relaxed the nonnegative constraint in the above Lagrange multiplier. However, we will show that the solution obtained will automatically be nonnegative mainly because of the logarithm term in the cost function. Taking the derivative of L with respect to θ_d and λ and setting them to zero, we have, respectively,

$$\frac{\partial L}{\partial \theta_d} = \nabla_{\theta_d} f(\theta_{d,\ell-1}) + \frac{1}{T_{d,\ell}} \left[\ln \frac{\theta_d}{\theta_{d,\ell-1}} \right] + \lambda \mathbb{1} = 0$$
$$\frac{\partial L}{\partial \lambda} = \mathbb{1}^T \theta_d - 1 = 0$$

which leads to

$$\theta_d = \frac{1}{\lambda} \theta_{d,\ell-1} \odot \exp\left(-T_{d,\ell} \cdot \nabla_{\theta_d} f(\theta_{d,\ell-1})\right)$$
$$\mathbb{1}^T \theta_d = 1$$

Solving the above two equations together, we obtain

$$\theta_d = \frac{1}{C_\theta} \theta_{d,\ell-1} \odot \exp\left(-T_{d,\ell} \cdot \nabla_{\theta_d} f(\theta_{d,\ell-1})\right) \tag{26}$$

where C_θ is a normalization factor such that $\theta_{d,\ell}$ adds up to one. Note that the above recursion can always guarantee non-negativity of the entries in the vector $\theta_{d,\ell}$ since we will always initialize the vector in the feasible region. Recall that $f(\theta_d)$ is the cost function on the right-hand side of (10), which is given by

$$f(\theta_d) = -x_d^T \ln(\Phi\theta_d) - (\alpha - \mathbb{1})^T \ln \theta_d$$

Therefore, the gradient of $f(\theta_d)$ can be computed as

$$\nabla_{\theta_d} f(\theta_d) = -\frac{x_d}{\Phi\theta_d} - \frac{\alpha - \mathbb{1}}{\theta_d} \tag{27}$$

Substituting the above gradient formula into (26), we obtain the desired result in (12).

C Implementation Details of the BP-sLDA

In this section, we describe the implementation details of the mirror-descent back propagation for the end-to-end learning of the supervised LDA model. Specifically, we will describe the details of the inference algorithm, and the model parameter estimation algorithm.

C.1 Inference algorithm: Mirror Descent

Let $f(\theta_d)$ denote the objective function in (12). As we discussed in the paper, we use recursion (12) to iteratively find the MAP estimate of θ_d given $w_{d,1:N}$, which we repeat below:

$$\theta_{d,\ell} = \frac{1}{C_\theta} \cdot \theta_{d,\ell-1} \odot \exp\left(T_{d,\ell} \left[\Phi^T \frac{x_d}{\Phi\theta_{d,\ell-1}} + \frac{\alpha - \mathbb{1}}{\theta_{d,\ell-1}} \right] \right), \ \ell = 1, \ldots, L, \quad \theta_{d,0} = \frac{1}{K}\mathbb{1} \tag{28}$$

The step-size $T_{d,\ell}$ in mirror descent can be chosen to be either constant, i.e., $T_{d,\ell} = T$, or adaptive over iterations ℓ and documents d. To adaptively determine the step-size, we can use line search procedure. A simple line search can be implemented as follows. For each document d:

- Initialization: $T_{d,0} = T_{d-1,L}/\eta$, where $0 < \eta < 1$ (e.g., $\eta = 0.5$).
- Repeat:
  - Update $\theta_{d,\ell}$ by (28) .

– Break if following condition holds:

$$f(\theta_{d,\ell}) \leq f(\theta_{d,\ell-1}) + [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T(\theta_{d,\ell} - \theta_{d,\ell-1}) + \frac{1}{2T_{d,\ell}}\Psi(\theta_{d,\ell}, \theta_{d,\ell-1}) \tag{29}$$

else: $T_{d,\ell} \leftarrow \eta \cdot T_{d,\ell}$

Moreover, $\Psi(\theta_{d,\ell}, \theta_{d,\ell-1})$ can also be replaced by the squared vector 1-norm:

$$f(\theta_{d,\ell}) \leq f(\theta_{d,\ell-1}) + [\nabla_{\theta_d} f(\theta_{d,\ell-1})]^T(\theta_{d,\ell} - \theta_{d,\ell-1}) + \frac{1}{2T_{d,\ell}}\|\theta_{d,\ell} - \theta_{d,\ell-1}\|_1^2 \tag{30}$$

The line search approach determines the step-sizes adaptively, automatically stabilizing the algorithm and making inference converge faster.

C.2 Parameter Estimation: Stochastic Gradient Descent with Back Propagation

We first rewrite the training cost (14) as

$$J(U, \Phi) = \sum_{d=1}^{D} Q_d(U, \Phi) \tag{31}$$

where $Q_d(\cdot)$ denotes the loss function at the d-th document, defined as

$$Q_d(U, \Phi) \triangleq -\frac{1}{D}\ln p(\Phi|\beta) - \ln p(y_d|\theta_{d,L}, U, \gamma) \tag{32}$$

Note that, we do not have constraint on the model parameter U. Therefore, to update U, we can directly use the standard mini-batch stochastic gradient descent algorithm. We randomly sample a mini-batch of documents, and then perform MAP inference of θ_d for each document in the mini-batch. And then, we compute the stochastic gradient of the loss function for each document, and use the averaged stochastic gradient to update U.

On the other hand, each column of the model parameter Φ is constrained to be on a $(V-1)$-dimension probability simplex, i.e, each element of Φ has to be nonnegative and each column sum up to one (i.e., Φ is a left-stochastic matrix). For this reason, we need to enforce the constraint on Φ. Recalling the definition of the Dirichlet smoothing $p(\Phi|\beta)$, we have

$$\begin{aligned} -\frac{1}{D}\ln p(\Phi|\beta) &= -\frac{1}{D}\ln\left(\left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^K \prod_{j=1}^{K}\prod_{v=1}^{V}\Phi_{vj}^{\beta-1}\right) \\ &= -\frac{1}{D}\sum_{j=1}^{K}\sum_{v=1}^{V}(\beta-1)\ln\Phi_{vj} + C \end{aligned} \tag{33}$$

Observe that expression (33) provides a natural log barrier for each element of Φ to enforce it to be nonnegative. Therefore, we can relax the nonnegative constraint on the elements of Φ and focus on enforcing the left-stochastic constraint. Let ϕ_j be the j-th column of Φ, we use the following algorithm to update the estimate of ϕ_j:

- Sample a mini-batch of documents.
- Perform MAP inference of y_d and θ_d using (12) for each document in the mini-batch.
- Compute the gradient $\partial Q_d/\partial\phi_j$ for each document d in the mini-batch and average them:

$$\Delta\phi_j = \frac{1}{D_b}\sum_{d\in\mathcal{D}_b}\frac{\partial Q_d}{\partial\phi_j}, \quad j = 1, \ldots, K$$

  where $\partial Q_d/\partial\phi_j$ is the j-th column of $\partial Q_d/\partial\Phi$, which can be computed according to the formula in Sec. D of this Appendix, $\mathcal{D}_b$ denotes the set of the documents in the mini-batch, and D_b is the number of documents in the mini-batch. The gradients are evaluated at $\phi_{j,t-1}$, the previous estimate of ϕ_j at time $t-1$.

- Set initial learning rate: $\mu_{\phi_j} = \mu_0$, $j = 1, \ldots, K$.
- For each $j = 1, \ldots, K$, repeat until all the elements of $\phi_{j,t}$ are nonnegative:
  - Update ϕ_j:

$$\phi_{j,t} = \Pi_{\{\phi:\mathbb{1}^T\phi=1\}}\left(\phi_{j,t-1} - \mu_{\phi_j} \cdot \Delta\phi_j\right) \tag{34}$$

    where $\Pi_{\{\phi:\mathbb{1}^T\phi=1\}}(\cdot)$ is the Euclidean projection operator onto the affine space: $\{\phi : \mathbb{1}^T\phi = 1\}$, which can be evaluated efficiently in closed-form:

$$\Pi_{\{\phi:\,\mathbb{1}^T\phi=1\}}(x) = \left(I - \frac{\mathbb{1}\mathbb{1}^T}{K}\right)x + \frac{1}{K}\mathbb{1}$$

  - Shrink the learning rate: $\mu_{\phi_j} = \eta\mu_{\phi_j}$, where $0 < \eta < 1$ (e.g., $\eta = 0.5$).

In the above iteration (34), we do not need to recompute the gradient $\Delta\phi_j$ but just iteratively shrink the learning rates until there is no violation of the nonnegativity constraint. This line search is only used to avoid the stochastic gradient descent from randomly moving ϕ_j into the nonnegative regime, where the log-barrier cannot push ϕ_j back to the positive region. Furthermore, we are allowing different columns of Φ to have different learning rates, which, from our observation in experiments, makes the training algorithm converge much faster than the uniform learning rate over all columns.

D Gradient Formula of BP-sLDA

In this section, we give the gradient formula for the supervised learning of BP-sLDA. To this end, we first rewrite the training cost (14) as

$$J(U, \Phi) = \sum_{d=1}^{D} Q_d(U, \Phi) \tag{35}$$

where $Q_d(\cdot)$ denotes the loss function at the d-th document, defined as

$$Q_d(U, \Phi) \triangleq -\frac{1}{D}\ln p(\Phi|\beta) - \ln p(y_d|\theta_{d,L}, U, \gamma) \tag{36}$$

The expressions for the two terms in (36) are given by

$$\begin{aligned} -\frac{1}{D}\ln p(\Phi|\beta) &= -\frac{1}{D}\ln\left(\left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^K \prod_{j=1}^{K}\prod_{v=1}^{V}\Phi_{vj}^{\beta-1}\right) \\ &= -\frac{1}{D}\sum_{j=1}^{K}\sum_{v=1}^{V}(\beta-1)\ln\Phi_{vj} + C \end{aligned} \tag{37}$$

$$\begin{aligned} -\ln p(y_d|\theta_{d,L}, U, \gamma) &= \begin{cases} -\sum_{j=1}^{V} y_{d,j}\ln\frac{\exp(\gamma\cdot p_{o,d,j})}{\sum_{m=1}^{V}\exp(\gamma\cdot p_{o,d,m})} & \text{classification} \\ \frac{1}{2\gamma}\|y_d - p_{o,d}\|_2^2 + C & \text{regression} \end{cases} \\ &= \begin{cases} -\sum_{j=1}^{V} y_{d,j}\gamma\cdot p_{o,d,j} + \ln\sum_{m=1}^{V}\exp(\gamma\cdot p_{o,d,m}) & \text{classification} \\ \frac{1}{2\gamma}\|y_d - p_{o,d}\|_2^2 + C & \text{regression} \end{cases} \end{aligned} \tag{38}$$

where C in the above expressions denotes a constant term that is independent of U and Φ, and

$$p_{o,d} \triangleq U\theta_{d,L} \tag{39}$$

In order to apply stochastic gradient descent to minimize (35), it suffices to evaluate the gradient of $Q_d(U, \Phi)$ with respect to U and Φ, which we now proceed to derive. Note that the choice of $p(y_d|\theta_{d,L}, U, \gamma)$ is not restricted to the above two options in our framework. Other forms could also be used and the corresponding gradient formula could also be derived. However, in this paper, we only list the gradient formula for these two classical choices.

D.1 Gradient with respect to U

First, we derive the gradient of $Q_d(\cdot)$ with respect U. Note that the only term in (36) depending on U is $\ln p(y_d|\theta_{d,L}, U, \gamma)$. Therefore, we have $\partial Q_d/\partial U = -\partial \ln p(y_d|\theta_{d,L}, U, \gamma)/\partial U$. Taking the gradient of (38) with respect to U and after some simple algebra, we get

$$\frac{\partial Q_d}{\partial U} = \begin{cases} -\gamma \cdot (y_d - \hat{y}_d)\theta_{d,L}^T & \text{classification} \\ -\frac{1}{\gamma} \cdot (y_d - \hat{y}_d)\theta_{d,L}^T & \text{regression} \end{cases} \tag{40}$$

where $\hat{y}_d$ is defined as

$$\hat{y}_d = \begin{cases} \sigma(\gamma \cdot p_{o,d}) & \text{classification} \\ p_{o,d} & \text{regression} \end{cases}$$

where $\sigma(\cdot)$ is the soft-max function:

$$\sigma(x) \triangleq \frac{x}{\sum_{m=1}^{V} \exp(x_m)}$$

D.2 Gradient with respect to Φ

In this subsection, we give the final expression for the gradient of Q_d with respect to Φ. The derivation can be found in Sec. D.3 of this Appendix.

$$\frac{\partial Q_d}{\partial \Phi} = -\frac{1}{D} \cdot \frac{\beta - 1}{\Phi} + \sum_{\ell=1}^{L} \frac{\partial Q_d}{\partial \Phi_\ell} \tag{41}$$

where $\partial Q_d/\partial \Phi_\ell$ is defined as

$$\frac{\partial Q_d}{\partial \Phi_\ell} = T_{d,\ell} \cdot \left\{ \frac{x_d}{\Phi\theta_{d,\ell-1}} (\theta_{d,\ell} \odot \xi_{d,\ell})^T - \left[\Phi(\theta_{d,\ell} \odot \xi_{d,\ell}) \odot \frac{x_d}{(\Phi\theta_{d,\ell-1})^2} \right] \theta_{d,\ell-1}^T \right\} \tag{42}$$

and $\xi_{d,\ell}$ is an intermediate error vector computed from the following backward recursion:

$$\xi_{d,\ell-1} = (I - \mathbb{1}\theta_{d,\ell-1}^T) \left\{ \frac{\theta_{d,\ell} \odot \xi_{d,\ell}}{\theta_{d,\ell-1}} - T_{d,\ell} \cdot \left[\Phi^T \text{diag}\left(\frac{x_d}{(\Phi\theta_{d,\ell-1})^2} \right) \Phi + \text{diag}\left(\frac{\alpha - \mathbb{1}}{\theta_{d,\ell-1}^2} \right) \right] (\theta_{d,\ell} \odot \xi_{d,\ell}) \right\} \tag{43}$$

which is initialized at

$$\xi_{L,t} = -(I - \mathbb{1}\theta_{d,L}^T) \cdot U^T \cdot \gamma(y_d - \hat{y}_d) \tag{44}$$

In the above back propagation formula, x_d and y_d are the input bag-of-words vector and the label. The quantities $\theta_{d,\ell}$ and $\hat{y}_d$ are obtained from the inference step, the MDA step-size $T_{d,\ell}$ is either set to be a constant (as a hyper-parameters) or determined by line-search in the inference step.

Similar to the inference iteration (12), the above gradients can be computed efficiently by exploiting the sparsity of the vector x_d. For example, only the elements at the nonzero positions of x_d need to be computed for $\Phi\theta_{d,\ell-1}$ and $\Phi(\theta_{d,\ell} \odot \xi_{d,\ell})$ since $\frac{x_d}{\Phi\theta_{d,\ell-1}}$ and $\frac{x_d}{(\Phi\theta_{d,\ell-1})^2}$ are known to be zero at these positions. Moreover, although $(\beta - 1)/\Phi$ is a dense matrix operation, it is the same within one mini-batch and can therefore be computed only once over each mini-batch, which can significantly reduce the amount of computation.

D.3 Derivation of the gradient with respect to Φ

In this subsection, we derive the gradient formula for Φ. Note from (36) that, there are two terms that depend on Φ, and

$$\frac{\partial Q_d}{\partial \Phi} = \frac{\partial}{\partial \Phi} \left(-\frac{1}{D} \ln p(\Phi|\beta) \right) + \frac{\partial}{\partial \Phi} \Big(-\ln p(y_d|\theta_{d,L}, U, \gamma) \Big) \tag{45}$$

The first term depends on Φ explicitly and its gradient can be evaluated direct as

$$\begin{aligned}\frac{\partial}{\partial\Phi}\left(-\frac{1}{D}\ln p(\Phi|\beta)\right) &= \frac{\partial}{\partial\Phi}\left(-\frac{1}{D}\sum_{j=1}^{K}\sum_{v=1}^{V}(\beta-1)\ln\Phi_{vj}\right)\\ &= -\frac{1}{D}\cdot\frac{\beta-1}{\Phi}\end{aligned} \tag{46}$$

The second term, however, depends on Φ implicitly through $\theta_{d,L}$. From Figure 2, we observe that $\theta_{d,L}$ not only depends on Φ explicitly (as indicated in the MDA block on the right-hand side of Figure 2) but also depends on Φ implicitly via $\theta_{d,L-1}$, which in turn depends on Φ both explicitly and implicitly (through $\theta_{d,L-2}$) and so on. That is, the dependency of the cost function on Φ is in a layered manner. For this reason, we need to apply chain rule to derive the its full gradient with respect to Φ, which we describe below.

First, as we discussed above, each MDA block in Figure 2 contains Φ, and $Q_d(U,\Phi)$ depends on the Φ appeared at different layers through $\theta_{d,L},\ldots,\theta_{d,1}$. If we denote these Φ at different layers as $\Phi_L,\ldots,\Phi_1$, and introduce an auxiliary function $Q_d(U,\Phi_1,\ldots,\Phi_L)$ to represent an artificial function, $-\ln p(y_d|\theta_{d,L},U,\gamma)$, with this "untied" Φ across layers in Figure 2, then the original $-\ln p(y_d|\theta_{d,L},U,\gamma)$ with "tied" Φ across layers can be written in the form of $Q_d(U,\Phi_1,\ldots,\Phi_L)$ as

$$-\ln p(y_d|\theta_{d,L},U,\gamma) = Q_d(U,\Phi,\ldots,\Phi) \tag{47}$$

For this reason, we can express the gradient of $-\ln p(y_d|\theta_{d,L},U,\gamma)$ with respect to Φ as

$$\frac{\partial}{\partial\Phi}\Big(-\ln p(y_d|\theta_{d,L},U,\gamma)\Big) = \sum_{\ell=1}^{L}\frac{\partial Q_d}{\partial\Phi_\ell} \tag{48}$$

where $\partial Q_d/\partial\Phi_\ell$ denotes the gradient of $Q(U,\Phi_1,\ldots,\Phi_L)$ with respect to Φ_ℓ evaluated at $\Phi_1=\Phi_2=\cdots=\Phi_L=\Phi$. Therefore, we only need to compute the gradient $\partial Q_d/\partial\Phi_\ell$.

For simplicity of notation, we drop the subscript of d in $\theta_{d,\ell}$ and define the following intermediate quantities:

$$\begin{aligned}z_\ell &= T_{d,\ell}\cdot\left[\Phi^T\frac{x_d}{\Phi\theta_{\ell-1}}+\frac{\alpha-\mathbb{1}}{\theta_{\ell-1}}\right]\\ p_\ell &= \theta_{\ell-1}\odot\exp(z_\ell)\end{aligned}$$

Then the MDA inference recursion (12) can be written in the following equivalent form:

$$z_\ell = T_{d,\ell}\cdot\left[\Phi^T\frac{x_d}{\Phi\theta_{\ell-1}}+\frac{\alpha-\mathbb{1}}{\theta_{\ell-1}}\right] \tag{49}$$

$$p_\ell = \theta_{\ell-1}\odot\exp(z_\ell) \tag{50}$$

$$\theta_\ell = \frac{p_\ell}{\mathbb{1}^T p_\ell} \tag{51}$$

To derive the gradient $\partial Q_d/\partial\Phi_\ell$, it suffices to derive $\frac{\partial Q}{\partial\Phi_{\ell,ji}}$. Note that

$$\frac{\partial Q_d}{\partial\Phi_{\ell,ji}} = \frac{\partial p_\ell^T}{\partial\Phi_{\ell,ji}}\cdot\frac{\partial Q_d}{\partial p_\ell} = \frac{\partial p_\ell^T}{\partial\Phi_{\ell,ji}}\cdot\delta_\ell \tag{52}$$

where

$$\delta_\ell \triangleq \frac{\partial Q_d}{\partial p_\ell} \tag{53}$$

is an intermediate quantities which follows a backward recursion to be derived later. To proceed, we need to derive $\partial p_\ell^T/\partial\Phi_{\ell,ji}$:

$$\frac{\partial p_\ell^T}{\partial\Phi_{\ell,ji}} = \theta_{\ell-1}^T\odot\frac{\partial\exp(z_\ell^T)}{\partial\Phi_{\ell,ji}}$$

$$
\begin{aligned}
&= \theta_{\ell-1}^T \odot \left[\frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} \cdot \mathrm{diag}\big(\exp(z_\ell)\big) \right] \\
&= \theta_{\ell-1}^T \odot \left[\frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} \odot \mathbb{1} \exp(z_\ell^T) \right] \\
&= \theta_{\ell-1}^T \odot \exp(z_\ell^T) \odot \frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} \\
&= p_\ell^T \odot \frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} \qquad (54)
\end{aligned}
$$

Then, we need to derive the expression for $\frac{\partial z_l^T}{\partial \Phi_{\ell,ji}}$:

$$
\begin{aligned}
\frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} &= T_{d,\ell} \cdot \left\{ \frac{\partial}{\partial \Phi_{\ell,ji}} \left(\frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \right) \cdot \Phi_\ell + \frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \cdot \frac{\Phi_\ell}{\Phi_{\ell,ji}} \right\} \\
&= T_{d,\ell} \cdot \left\{ \frac{\partial}{\partial \Phi_{\ell,ji}} \left(\frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \right) \cdot \Phi_\ell + \frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \cdot E_{ji} \right\} \\
&= T_{d,\ell} \cdot \left\{ -\frac{\partial \theta_{\ell-1}^T \Phi_\ell^T}{\partial \Phi_{\ell,ji}} \cdot \mathrm{diag} \left(\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right) \cdot \Phi_\ell + \frac{x_d^T}{\theta_{\ell-1}^T \Phi_l^T} \cdot E_{ji} \right\} \\
&= T_{d,\ell} \cdot \left\{ -\theta_{\ell-1}^T E_{ij} \cdot \mathrm{diag} \left(\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right) \cdot \Phi_\ell + \frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \cdot E_{ji} \right\} \\
&= T_{d,\ell} \cdot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j e_j^T \Phi_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j e_i^T \right\} \qquad (55)
\end{aligned}
$$

where e_i denotes a one-hot vector with the i-th element being one and all other element equal to zero, and E_{ji} denotes a matrix whose (j,i)-th element is one and all other elements are zero. Substituting the above expression into (54), we obtain

$$
\begin{aligned}
\frac{\partial p_\ell^T}{\partial \Phi_{\ell,ji}} &= p_\ell^T \odot \frac{\partial z_\ell^T}{\partial \Phi_{\ell,ji}} \\
&= T_{d,\ell} \cdot p_\ell^T \odot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j e_j^T \Phi_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j e_i^T \right\} \qquad (56)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\frac{\partial Q_d}{\partial \Phi_{\ell,ji}} &= \frac{\partial p_\ell^T}{\partial \Phi_{\ell,ji}} \cdot \delta_\ell \\
&= T_{d,\ell} \cdot p_\ell \odot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j e_j^T \Phi_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j e_i^T \right\} \delta_\ell \\
&= T_{d,\ell} \cdot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j \left(p_\ell \odot e_j^T \Phi_\ell \right) \delta_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j (p_\ell \odot e_i^T) \delta_\ell \right\} \\
&= T_{d,\ell} \cdot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j \left(p_\ell \odot e_j^T \Phi_\ell \right) \delta_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j [p_\ell]_i \cdot [\delta_\ell]_i \right\} \\
&= T_{d,\ell} \cdot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j \left(e_j^T \Phi_\ell \mathrm{diag}(p_\ell) \right) \delta_\ell + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j [p_\ell]_i \cdot [\delta_\ell]_i \right\} \\
&= T_{d,\ell} \cdot \left\{ -[\theta_l]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j e_j^T \Phi_\ell (p_{l-1} \odot \delta_\ell) + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j [p_\ell]_i \cdot [\delta_\ell]_i \right\} \\
&= T_{d,\ell} \cdot \left\{ -[\theta_{\ell-1}]_i \left[\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right]_j [\Phi_\ell (p_\ell \odot \delta_\ell)]_j + \left[\frac{x_d}{\Phi_\ell \theta_{\ell-1}} \right]_j [p_\ell]_i \cdot [\delta_\ell]_i \right\} \qquad (57)
\end{aligned}
$$

Writing the above expressions into matrix form (derivative with respect Φ_ℓ), we obtain:

$$\frac{\partial Q_d}{\partial \Phi_\ell} = T_{d,\ell} \cdot \left\{ \frac{x_d}{\Phi_\ell \theta_{\ell-1}} (p_\ell \odot \delta_\ell)^T - \left[\Phi_\ell (p_\ell \odot \delta_\ell) \odot \frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right] \theta_{\ell-1}^T \right\} \tag{58}$$

Now we need to derive the recursion for computing δ_ℓ. By the definition of δ_ℓ in (53), we have

$$\begin{aligned} \delta_{\ell-1} &\triangleq \frac{\partial Q_d}{\partial p_{\ell-1}} \\ &= \frac{\partial \theta_{\ell-1}^T}{\partial p_{\ell-1}} \cdot \frac{\partial p_\ell^T}{\partial \theta_{\ell-1}} \cdot \frac{\partial Q_d}{\partial p_\ell} \\ &= \frac{\partial \theta_{\ell-1}^T}{\partial p_{\ell-1}} \cdot \frac{\partial p_\ell^T}{\partial \theta_{\ell-1}} \cdot \delta_\ell \end{aligned} \tag{59}$$

To continue, we have to evaluate $\frac{\partial \theta_{\ell-1}^T}{\partial p_{\ell-1}}$ and $\frac{\partial p_\ell^T}{\partial \theta_{\ell-1}}$. By (49)–(51), we have

$$\begin{aligned} \frac{\partial p_\ell^T}{\partial \theta_{\ell-1}} &= \frac{\partial \theta_{\ell-1}^T}{\partial \theta_{\ell-1}} \odot \mathbb{1} \exp(z_\ell^T) + \mathbb{1}\theta_{\ell-1}^T \odot \frac{\partial \exp(z_\ell^T)}{\partial \theta_{\ell-1}} \\ &= I \odot [\mathbb{1} \exp(z_\ell^T)] + \mathbb{1}\theta_{\ell-1}^T \odot \left[\frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} \cdot \frac{\partial e_\ell^T}{\partial z_\ell} \right] \\ &= \operatorname{diag}\big(\exp(z_\ell)\big) + \mathbb{1}\theta_{\ell-1}^T \odot \left[\frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} \cdot \operatorname{diag}\big(\exp(z_\ell)\big) \right] \\ &= \operatorname{diag}\big(\exp(z_\ell)\big) + \mathbb{1}\theta_{\ell-1}^T \odot \left[\frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} \odot \mathbb{1} \exp(z_\ell^T) \right] \\ &= \operatorname{diag}\big(\exp(z_\ell)\big) + \mathbb{1} \left[\theta_{\ell-1}^T \odot \exp(z_\ell^T) \right] \odot \frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} \\ &= \operatorname{diag}\big(\exp(z_\ell)\big) + \mathbb{1} p_\ell^T \odot \frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} \end{aligned} \tag{60}$$

To proceed, we need to derive the expression for $\frac{\partial z_\ell^T}{\partial \theta_{\ell-1}}$:

$$\begin{aligned} \frac{\partial z_\ell^T}{\partial \theta_{\ell-1}} &= T_{d,\ell} \cdot \left\{ \frac{\partial}{\partial \theta_{\ell-1}} \left(\frac{x_d^T}{\theta_{\ell-1}^T \Phi_\ell^T} \right) \Phi_\ell + \frac{\partial}{\partial \theta_{\ell-1}} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}} \right)^T \right\} \\ &= T_{d,\ell} \cdot \left\{ -\frac{\partial \theta_{\ell-1}^T \Phi_\ell^T}{\partial \theta_{\ell-1}} \cdot \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell^T \theta_{\ell-1})^2} \right) \Phi_\ell - \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right\} \\ &= T_{d,\ell} \cdot \left\{ -\Phi_\ell^T \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell^T \theta_{\ell-1})^2} \right) \Phi_\ell - \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right\} \\ &= -T_{d,\ell} \cdot \left\{ \Phi_\ell^T \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell^T \theta_{\ell-1})^2} \right) \Phi_\ell + \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right\} \end{aligned} \tag{61}$$

Substituting the above expression into (60), we get the expression for $\frac{\partial p_\ell^T}{\partial \theta_{\ell-1}}$:

$$\begin{aligned} \frac{\partial p_\ell^T}{\partial \theta_{\ell-1}} &= \operatorname{diag} \left\{ \exp \left(T_{d,\ell} \left[\Phi_\ell^T \frac{x_d}{\Phi_\ell \theta_{\ell-1}} + \frac{\alpha - \mathbb{1}}{\theta_{\ell-1}} \right] \right) \right\} \\ &\quad - T_{d,\ell} \cdot (\mathbb{1} p_\ell^T) \odot \left[\Phi_\ell^T \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right) \Phi_\ell + \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right] \\ &= \operatorname{diag} \left(\frac{p_\ell}{\theta_{\ell-1}} \right) - T_{d,\ell} \cdot (\mathbb{1} p_\ell^T) \odot \left[\Phi_\ell^T \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right) \Phi_\ell + \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right] \\ &= \left\{ \operatorname{diag} \left(\frac{1}{\theta_{\ell-1}} \right) - T_{d,\ell} \cdot \left[\Phi_\ell^T \operatorname{diag} \left(\frac{x_d}{(\Phi_\ell \theta_{\ell-1})^2} \right) \Phi_\ell + \operatorname{diag} \left(\frac{\alpha - \mathbb{1}}{\theta_{\ell-1}^2} \right) \right] \right\} \operatorname{diag}(p_\ell) \end{aligned} \tag{62}$$

To complete the derivation of the recursion (59), we need to derive $\frac{\partial \theta_{\ell-1}^T}{\partial p_{\ell-1,t}}$, which is given by

$$\frac{\partial \theta_{\ell-1}^T}{\partial p_{\ell-1}} = \frac{\partial p_{\ell-1}^T}{\partial p_{\ell-1}} \cdot \frac{1}{\mathbb{1}^T p_{\ell-1}} + \frac{\partial}{\partial p_{\ell-1}} \left(\frac{1}{\mathbb{1}^T p_{\ell-1}} \right) p_{\ell-1}^T = \frac{I - \mathbb{1}\theta_{\ell-1}^T}{\mathbb{1}^T p_{\ell-1}} \tag{63}$$

Expressions (59), (62) and (63) provide the complete backward recursion for δ_ℓ, which starts from $\ell = L$ and ends at $\ell = 2$. Finally, to initialize this backward recursion, we need to derive the expression for δ_L. By its definition, we have

$$\begin{aligned} \delta_L &\triangleq \frac{\partial Q_d}{\partial p_L} \\ &= \frac{\partial \theta_L^T}{\partial p_L} \cdot \frac{\partial p_{o,d}^T}{\partial \theta_L} \cdot \frac{\partial Q_d}{\partial p_{o,d}} \\ &= \frac{\partial \theta_L^T}{\partial p_L} \cdot U^T \cdot \frac{\partial Q_d}{\partial p_{o,d}} \\ &= \frac{1}{\mathbb{1}^T p_L} (I - \mathbb{1}\theta_L^T) \cdot U^T \cdot \frac{\partial Q_d}{\partial p_{o,d}} \end{aligned} \tag{64}$$

where in the last step we substituted (63). By (47) and(38), we have

$$\begin{aligned} \frac{\partial Q_d}{\partial p_{o,d}} &= \frac{\partial}{\partial p_{o,d}} \Big(- \ln p(y_d | \theta_{d,L}, U, \gamma) \Big) \\ &= \begin{cases} -\gamma \cdot (y_d - \hat{y}_d) & \text{classification} \\ -\frac{1}{\gamma} \cdot (y_d - \hat{y}_d) & \text{regression} \end{cases} \end{aligned} \tag{65}$$

Therefore,

$$\delta_L = \begin{cases} -\dfrac{1}{\mathbb{1}^T p_L} (I - \mathbb{1}\theta_L^T) \cdot U^T \cdot \gamma \cdot (y_d - \hat{y}_d) & \text{classification} \\ -\dfrac{1}{\mathbb{1}^T p_L} (I - \mathbb{1}\theta_L^T) \cdot U^T \cdot \dfrac{1}{\gamma} \cdot (y_d - \hat{y}_d) & \text{regression} \end{cases} \tag{66}$$

As a final remark, we found in practical implementation that p_ℓ could be very large while δ_ℓ could be small, which leads to potential numerical instability. To address this issue, we introduce the following new variable:

$$\xi_{d,\ell} \triangleq \mathbb{1}^T p_\ell \cdot \delta_\ell \tag{67}$$

Then, the quantities p_ℓ and δ_ℓ can be replaced with one variable $\xi_{d,\ell}$, and the backward recursion of δ_ℓ can also be replaced with the backward recursion of $\xi_{d,\ell}$. With some simple algebra, we obtain the final gradient expression for Φ in Appendix D.2.

E Gradient Formula of BP-LDA

The unsupervised learning problem (4) can be rewritten, equivalently, as minimizing the following cost function:

$$J(\tilde{\Phi}) = \sum_{d=1}^{D} Q_d(\tilde{\Phi}) \tag{68}$$

where $Q_d(\tilde{\Phi})$ is the loss function defined as

$$Q_d(\tilde{\Phi}) = -\frac{1}{D} \ln p(\tilde{\Phi}|\beta) - \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha) \tag{69}$$

Taking the gradient of both sides of (69), we obtain

$$\frac{\partial Q_d}{\partial \tilde{\Phi}} = \frac{\partial}{\partial \tilde{\Phi}} \left(-\frac{1}{D} \ln p(\tilde{\Phi}|\beta) \right) + \frac{\partial}{\partial \tilde{\Phi}} \Big(- \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha) \Big) \tag{70}$$

The first term in (70) has already been derived in (46):

$$\frac{\partial}{\partial \tilde{\Phi}} \ln p(\tilde{\Phi}|\beta) = \frac{\beta - 1}{\tilde{\Phi}} \tag{71}$$

where $\frac{\beta-1}{\tilde{\Phi}}$ denotes elementwise division of the scalar $\beta - 1$ by the matrix $\tilde{\Phi}$. We now proceed to derive the second term in (70).

$$\begin{aligned}
\frac{\partial}{\partial \tilde{\Phi}} \ln p(w_{d,1:N}|\tilde{\Phi}, \alpha) &= \frac{1}{p(w_{d,1:N}|\tilde{\Phi}, \alpha)} \cdot \frac{\partial}{\partial \tilde{\Phi}} p(w_{d,1:N}|\tilde{\Phi}, \alpha) \\
&= \frac{1}{p(w_{d,1:N}|\tilde{\Phi}, \alpha)} \cdot \int_{\theta_d} \left[\frac{\partial}{\partial \tilde{\Phi}} p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \right] d\theta_d \\
&= \frac{1}{p(w_{d,1:N}|\tilde{\Phi}, \alpha)} \cdot \int_{\theta_d} \left[\frac{\partial}{\partial \tilde{\Phi}} \ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \right] \cdot p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) d\theta_d \\
&= \int_{\theta_d} \left[\frac{\partial}{\partial \tilde{\Phi}} \ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \right] \cdot \frac{p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha)}{p(w_{d,1:N}|\tilde{\Phi}, \alpha)} d\theta_d \\
&= \int_{\theta_d} \left[\frac{\partial}{\partial \tilde{\Phi}} \ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \right] \cdot p(\theta_d|w_{d,1:N}, \tilde{\Phi}, \alpha) d\theta_d \\
&= \mathbb{E}_{\theta_d|w_{d,1:N}} \left[\frac{\partial}{\partial \tilde{\Phi}} \ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \right]
\end{aligned} \tag{72}$$

To continue, we now derive the expression for the gradient inside the expectation term in (72). By (9) and the definition of Dirichlet distribution $p(\theta_d|\alpha)$, we can write $\ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha)$ as

$$\begin{aligned}
\ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) &= \ln p(w_{d,1:N}, \theta_d|\tilde{\Phi}, \alpha) \\
&= \ln p(w_{d,1:N}|\theta_d, \tilde{\Phi}) + \ln p(\theta_d|\alpha) \\
&= \sum_{j=1}^{K} (\alpha_j - 1) \ln \theta_{d,j} + \ln \Gamma\Big(\sum_{j=1}^{K} \alpha_j \Big) - \sum_{j=1}^{K} \ln \Gamma(\alpha_j) \\
&\quad + \sum_{v=1}^{V} x_{d,v} \ln \Big(\sum_{j=1}^{K} \theta_{d,j} \tilde{\Phi}_{vj} \Big) \\
&= x_d^T \ln(\tilde{\Phi}\theta_d) + (\alpha - 1)^T \ln \theta_d + \ln(\mathbb{1}^T \alpha) - \mathbb{1}^T \ln \Gamma(\alpha)
\end{aligned} \tag{73}$$

Taking the gradient of the above expression with respect to $\tilde{\Phi}$, we obtain the gradient formula.