Topic Analysis for Text with Side Data

Anonymous ACL submission

Abstract

Although latent factor models (e.g., matrix factorization) perform well in predictions, they face challenges such as cold-start, lack of transparency, and suboptimal recommendations. In this paper, we leverage text with side data to address these issues. We propose a hybrid generative probabilistic model that integrates a neural network with a latent topic model within a four-level hierarchical Bayesian framework. Here, each document is a finite mixture over topics, each topic is an infinite mixture over topic probabilities, and each topic probability is a finite mixture over side data. The neural network produces an overview distribution of the side data, which serves as the LDA prior to improve topic grouping. Our experiments on various datasets show that the model outperforms standard LDA and Dirichlet-multinomial regression (DMR) in topic grouping, model perplexity, classification, and comment generation.

1 Introduction

As vast amounts of digital text—from news articles to blogs and web pages—are increasingly stored, discovering their underlying topics becomes challenging. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) has recently gained much popularity for its simplicity and its ability to project documents into a low-dimensional semantic space. However, modern text often comes with additional side data such as customer ratings, labels, or loyalty information, which can enhance topic discovery. Existing models that incorporate side data fall into two categories: (1) downstream models, which generate text and side data simultaneously, and (2) upstream models, which condition text generation on side data. Our model extends the upstream approach by using deep neural networks to capture more complex interactions between side data and text than models like DMR (Mimno and McCallum, 2008).

In this paper, we introduce hybrid neural network LDA (nnLDA), an LDA-style topic model that integrates a neural network with LDA. Unlike standard LDA—which uses a fixed prior—nnLDA generates a topic prior from side data via a neural network, making the prior sample-specific. This design allows the model to capture both the main text content and additional, non-dominant patterns from side data. The document-topic distribution is modeled as a mixture of feature-specific distributions, and the neural network parameters and topic-word distributions are jointly optimized using a stochastic EM algorithm. In the E-step, the optimal word and topic groups are identified, while the M-step updates the neural network parameters and topic-word distributions.

We not only propose a more general model, nnLDA, but also present a complete technical proof confirming that nnLDA performs at least as well as plain LDA in terms of log likelihood. Furthermore, we provide an efficient variational EM algorithm for nnLDA. Lastly, we demonstrate our approach on a few real-world datasets. In summary, we make the following contributions.

- We provide a new topic model for text datasets with side data.
- We prove that the lower bound of log likelihood of nnLDA is greater than or equal to the lower bound of log likelihood of LDA for any dataset.
- We provide an efficient variational EM algorithm for nnLDA.
- We present numerical results showing that nnLDA outperforms LDA and DMR in terms of topic grouping, model perplexity, classification and text generation.

2 Related Work

There are a large amount of extensions of the plain LDA model, however, a full retrospection of this immense literature exceeds the scope of this work. In this section, we state several kinds of variations of LDA which are most related to our new model and interpret the relationships among them.

LDA: Plain LDA has been widely used in text (Blei et al., 2003; Wang et al., 2020), image (Li and Perona, 2005), and network analysis (Airoldi et al., 2008) for its simplicity, low-dimensional representations, and coherent topics. However, modifying LDA typically requires re-deriving inference algorithms. To address this, Srivastava and Sutton (Srivastava and Sutton, 2017) proposed Neural Variational LDA (NVLDA) using a Logistic-Normal prior, while RollingLDA (Rieger et al., 2021) enables incremental updates. Additionally, Optimized LDA (OLDA) (Haritha and Shanmugavadivu, 2024) applies hyperparameter tuning to enhance topic extraction. Despite its success, LDA relies solely on bag-of-words representations, overlooking valuable side information. In contrast, nnLDA leverages a neural network to integrate side data, capturing secondary, non-dominant, and more salient patterns that can better inform topic inference.

Downstream Topic Models: Downstream models generate text and side data jointly from latent topics by associating each topic with distributions over both words and side data, optimizing their joint likelihood. Examples include Corr-LDA (Blei and Jordan, 2003), the mixed-membership model for authorship (Erosheva et al., 2004), Group-Topic model (Wang et al., 2005), TOT (Wang and McCallum, 2006), MedLDA (Zhu et al., 2012), and TUM (Jiang et al., 2013). For instance, TUM models query logs by capturing separate distributions for query terms and URLs, making it computationally demanding due to its distinct generative processes. Another flexible approach is supervised LDA (sLDA) (Blei and McAuliffe, 2007) (and its variants (Wang et al., 2009; Wang and McCallum, 2006)), which maximizes the joint likelihood of text and side data (e.g., customer ratings) via a GLM with a specified link and dispersion function. However, this requirement limits sLDA to a small set of side data vectors. In contrast, our model circumvents these limitations by adopting an entirely different approach.

Upstream Topic Model: Downstream models jointly generate text and side data from latent topics by maximizing their joint likelihood. Examples include Corr-LDA (Blei and Jordan, 2003), the mixed-membership model for authorship (Erosheva et al., 2004), Group-Topic models (Wang et al., 2005), TOT (Wang and McCallum, 2006), MedLDA (Zhu et al., 2012), and TUM (Jiang et al., 2013)—the latter separately modeling query terms and URLs, which increases computational cost.

In contrast, upstream models condition text generation on observed side data and maximize the conditional likelihood. For instance, DiscLDA (Lacoste-Julien et al., 2008), scene understanding models (Sudderth et al., 2005), and the author-topic model (Rosen-Zvi et al., 2004) generate words by first selecting an author and then sampling a topic from that author's distribution. Although some extensions (Rosen-Zvi et al., 2004; McCallum et al., 2007; Dietz et al., 2007) allow mixed topics per document, they typically use only ratings or labels and cannot handle multiple modalities simultaneously.

While earlier upstream methods project side data onto the topic prior using fixed operations (e.g., the dot product in DMR (Mimno and McCallum, 2008) and collective supervision (Benton et al., 2016)), nnLDA employs a neural network to learn an adaptive mapping. By dynamically generating a sample-specific prior from diverse side data, nnLDA accommodates both categorical and continuous modalities, thereby enhancing topic inference and overall performance.

3 Algorithm

We first present notation and the setting. We use the language of text collections throughout the paper, referring to terminologies such as "words," "documents" and "corpus" since it makes the concepts more intuitive to understand. In general, similar to plain LDA, nnLDA is not restricted to text datasets, and can also be applied on other kinds of datasets, i.e. image datasets.

- A *word*, defined as an item from a vocabulary indexed by $\{1, \cdots, V\}$, is applied one-hot encoding. More precisely, using superscripts to denote components, the v'th word in the vocabulary is represented by a V-vector w such that $w^v = 1$ and $w^u = 0$ for all $u \neq v$.
- A *document* is a set of N words denoted by $d = \mathbf{w} = \{w_1, w_2, \cdots, w_N\}$ if it only contains textual data. Similarly, if a doc-

ument contains q different kinds of side data together with the aforementioned textual data, we denote it by $d = (\mathbf{w}, \mathbf{s}) = (\{w_1, w_2, \cdots, w_N\}, (s_1, s_2, \cdots, s_q))$ where $\mathbf{s} \in \mathrm{R}^q$.

- A *corpus* is a collection of M documents denoted by $D = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_M\}$ for textual only documents and $(D, S) = \{(\mathbf{w}_1, \mathbf{s}_1), (\mathbf{w}_2, \mathbf{s}_2), \cdots, (\mathbf{w}_M, \mathbf{s}_M)\}$ for documents containing both side and textual data.

The main goal of nnLDA is to find a probabilistic model of a corpus that, by involving high-level summarization from side data, not only assigns high probability to documents in this corpus but also assigns high probability to other similar documents based on side data.

3.1 Generative Model

We propose the nnLDA model to explain the generative process of a document d with textual data $\mathbf{w}$ (containing N words) and side data (structural data) $\mathbf{s}$, the steps of which can be summarized as follows.

Algorithm 1 Generative Process of nnLDA

1: Choose $N \sim \text{Poisson}(\xi)$
2: Choose $\mathbf{s} \sim \mathcal{N}(\mu, \sigma^2 I)$
3: Choose $\alpha_d \leftarrow g(\gamma; \mathbf{s})$
4: Choose $\theta \sim \text{Dir}(\alpha_d)$
5: **for** each of the N words w_n **do**
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Notation "Poisson," "Dir" represents the Poisson and Dirichlet distribution, respectively. In step 3, g refers to a parametric model to generate α. In summary, the model has two trainable parameters: γ, the parameters of g for side info $\mathbf{s}$; β, the topic-word distribution. In the meanwhile, there are three hyper parameters: μ and σ^2, the mean and the variance of the probability distribution for side data $\mathbf{s}$; and K, which does not explicitly appear in the generative process, the number of topics.

Step 1 is independent of the remaining steps, which determines the number of words in the document. Then, for each document, step 2 provides a representation of side data $\mathbf{s}$ by using a normal distribution with mean μ and variance σ^2. Then, applying a model with input $\mathbf{s}$ in step 3 provides the prior α_d for the Dirichlet distribution. In step 3, the model $g(\gamma; \cdot)$ employed is a deep neural network, and we do not specify the architecture of the deep neural network in this study since different kinds of side data may inquire different deep neural networks. We leave the freedom of selecting the architecture of the deep neural network to the user. Next, the random parameter of a multinomial distribution over topics, θ, is generated by the Dirichlet distribution. Finally, for the n'th word in the document, step 5(a) first selects a topic z_n among the K different topics by the multinomial distribution with parameter θ, and then step 5(b) generates a word w_n based on the topic-word distribution β specific to topic z_n. Step 5 follows standard LDA.

By incorporating a neural network $g(\gamma; \cdot)$ to generate a document-specific Dirichlet prior from side data, nnLDA guarantees a likelihood that is at least as high as that of standard LDA. Under the assumption that $g(\gamma; \cdot)$ has finite sample expressivity (see Definition 1 in Appendix A), Theorem 1 shows that the optimized likelihood of nnLDA meets or exceeds that of LDA. Furthermore, if the side data positively influences the document generation process quantified by a constant $C > 1$ then the improvement in likelihood is bounded below by $C - 1$ (Theorem 2). These results establish a strong theoretical foundation for the enhanced performance of nnLDA over traditional LDA.

3.2 Variational Inference with EM Algorithm

We train the nnLDA model using a stochastic EM sampling scheme, in which we alternate between sampling topic assignments from the current prior distribution conditioned on the observed words and side data, and optimizing the parameters given the topic assignments.

Details are similar to those in (Blei et al., 2003). In this section, instead of showing all the details, we point out the differences from the derivation of plain LDA. By applying the Jensen's inequality and KL divergence between the variational posterior probability and the true posterior probability, which is a formally stated technique in (Blei et al., 2003), a lower bound of log likelihood reads

$$\begin{aligned} &L(\xi, \phi; \gamma, \beta) \\ &= \mathbb{E}_q\left[\log p(\theta \mid g(\gamma; \mathbf{s}))\right] + \mathbb{E}_q\left[\log p(z \mid \theta)\right] \\ &+ \mathbb{E}_q\left[\log p(\mathbf{w} \mid z, \beta)\right] - \mathbb{E}_q\left[\log q(\theta)\right] - \mathbb{E}_q\left[\log q(z)\right], \end{aligned} \tag{1}$$

where ξ, ϕ are variational parameters of θ and z, respectively, and $q(\cdot)$ represents the variational distribution. Then, the iterative algorithm is

1. (E-step) For each document, find the optimizing values of the variational parameters ξ and ϕ of z and θ, respectively.
2. (M-step) Maximize the resulting lower bound of log likelihood with respect to the model parameters γ and β.

Algorithm 2 E-step of hybrid neural network LDA

Initialize:
  $\phi_{ni}^0 \leftarrow \frac{1}{K}$ for all i and n
  $\xi_i \leftarrow [g(\gamma; \mathbf{s})]_i + \frac{N}{K}$ for all i
for t = $0, 1, 2, \cdots$ **do**
  for $n = 1, 2, \cdots, N$, **do**
    for $i = 1, 2, \cdots, K$, **do**
      $\phi_{ni}^{t+1} = \beta_{iw_n} \exp(\Psi(\xi_i^t))$
    end for
    Normalize ϕ_n^{t+1} to sum to 1
  end for
  $\xi^{t+1} = g(\gamma; \mathbf{s}) + \sum_{n=1}^{N} \phi_n^{t+1}$
end for

The E-step is exhibited in Algorithm 2, where Ψ is the digamma function, the first derivative of the log Gamma function. The variational parameters are set separately for each document, similar to the E-step in (Blei et al., 2003), but replacing the prior α with $g(\gamma; \mathbf{s})$. We run the E-step until it converges for each document.

The M-step is finding a maximum likelihood estimation with expected sufficient statistics for each document under the approximate posterior parameters ξ and ϕ, which are computed in the E-step. Likewise, since the log-likelihood objective related to β does not involve $g(\gamma; \mathbf{s})$, we are allowed to directly borrow the update rule of β from (Blei et al., 2003), which is

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N} \phi_{dni}^* w_{dn}^j.$$

In contrast, for the neural network parameter γ, we resort to log likelihood objective related to γ as follows,

$$L_{[\gamma]} = \sum_{d=1}^{M} \left(\log \Gamma(\sum_{j=1}^{K} [g(\gamma; \mathbf{s}_d)]_j) - \sum_{i=1}^{K} \log \Gamma([g(\gamma; \mathbf{s}_d)]_i) + \sum_{i=1}^{K} \left(([g(\gamma; \mathbf{s}_d)]_i - 1) \left(\Psi(\xi_{di}) - \Psi(\sum_{j=1}^{K} \xi_{dj}) \right) \right) \right),$$

where M is the number of documents in the corpus, and Ψ is the digamma function, the first derivative of the log Gamma function. Then, applying the backpropagation approach provides the derivative and the update rule for parameter γ.

4 Experimental Study

In this section, we compare the nnLDA model with standard LDA and the DMR model introduced in (Blei et al., 2003) and (Mimno and McCallum, 2008), respectively. We conduct experiments on four different-size datasets among which one is a synthetic dataset and the remaining three are real-world datasets. For these datasets, we study the performance of topic grouping, perplexity, classification and comment generation for nnLDA, plain LDA and DMR models. For each of the tasks, some datasets are not eligible to be examined due to lack of information. The synthetic dataset is publically available at https://github.com/biyifang/nnLDA/blob/main/syn_file.csv while the real-world datasets are proprietary.

4.1 Datasets and Training Details

The first dataset is a synthetic set of 2,000 samples. Each sample contains customer feedback regarding a purchase along with product characteristics. There are two categories: product (TV or burger) and description (price or quality). We assign a bag of words to each product–description combination as shown in Table 1. A category combination is randomly selected from the four, and a comment is generated by randomly choosing between one and five words (averaging 2.97 words) from the corresponding bag.

Category combination	Bag of words
(burger, price)	value, pricey, ouch, steep, cheap, value, reason, accept, unreason, unacceptable
(burger, quality)	nasty, fantastic, delicious, tasty, juicy, unreason, unacceptable, reason, accept, fresh
(TV, price)	promotion, affordable, value, increase, expensive, tasty, economical, fancy, okay
(TV, quality)	fabulous, fantastic, promising, sharp, large, clear, eco friendly, fresh, pixilated

Table 1: Synthetic Dataset

The second dataset, PTS, is a real-world set of 795 samples. Each sample includes a customer's short feedback and rating on a purchase, along with

product characteristics. The side data for nnLDA corresponds to sectors—a generalized product category. The shortest comment contains 1 word and the longest 49 words, with an average of 10.6 words per comment. For example, a customer in the Baby sector leaves a comment "Cheap& Soft" with a rating of 3.

The third dataset WIP is a medium-size dataset with 3,451 samples. Each sample contains a customer's short feedback and rating with respect to his or her purchase along with the characteristics of the product. The sector attribution is again side data when training models with one feature. The other attribution counted for models with two features is channel. The most concrete comment in the dataset has 138 words, while the briefest comment has only 1 word. In the meanwhile, the average length of the comments in the dataset is 8.9 words.

The last dataset DCL is another medium-size dataset of 5,427 samples. Different from the PTS and WIP datasets, each sample in DCL contains a customer's long feedback and rating with respect to his or her purchase along with the characteristics of the product. Additionally, the side data selected for nnLDA corresponds to groups of products. The smallest number of words for a comment in this dataset is 1, while the largest is 988. Overall, the average length of the comments is 61.7 words. A short sample comment is "quick points that will be all that matters to a buyer wanting accurate metrics to buy by tinny sound but plenty of audio hookups."

Dataset	Topic grouping	Perplexity	Classification	Comment Generation
Synthetic Dataset	Yes	No	No	No
PTS	No	Yes	Yes	Yes
WIP	No	Yes	Yes	Yes
DCL	No	Yes	Yes	No

Table 2: Tasks of Interest

Due to incomplete information in some datasets, we evaluate only selected tasks for each. For the topic grouping task, we assess nnLDA, plain LDA, and DMR on their ability to correctly cluster comments experiments are conducted only on the synthetic dataset where topic groups are well-defined. For perplexity, we compute the logarithm of the perplexity over all words, but do not evaluate this on the synthetic dataset (since its true number of topic groups is known). For classification, we use the probability vectors produced by the topic models to predict comment ratings. Finally, for comment generation, we test performance on the two smallest real-world datasets to examine behavior with limited samples. Table 2 summarizes the tasks evaluated for each dataset.

For all of these datasets, we employ a two-layer fully connected neural network as $g(\gamma; \cdot)$ in nnLDA. Furthermore, we set the number of neurons to be 20 in the first layer, the number of neurons of the second layer to be the number of topic groups assigned in the beginning and the batch size to be 64. All features of the side data are categorical and are one-hot encoded. Additionally, all weights in $g(\gamma; \cdot)$ are initialized by Kaiming Initialization (He et al., 2015). We apply the ADAM algorithm with the learning rate of 0.001 and weight decay being 0.1. Meanwhile, we train all the models using EM with exactly the same stopping criteria of stopping E-step and M-step when the average change over the whole training dataset in the expected log likelihood becomes less than 0.01%. We vary the number of topic groups from 4 to 30. For DMR, we use the same values for the parameters as those in (Mimno and McCallum, 2008). All the algorithms are implemented in Python with Pytorch and trained on a single GPU card.

4.2 Experimental Results

In this section, we present all the results based on the tasks of interest.

Overall, nnLDA outperforms plain LDA and DMR in all datasets in terms of topic grouping, classification, perplexity and comment generation. Meanwhile, based on the fact that the last two datasets have many more words and more intrinsic concepts in their comments when compared to the first three datasets, nnLDA exceeds the performance of plain LDA and DMR dramatically when a document contains several topics or it is more comprehensive.

4.2.1 Topic Grouping

Table 3 shows the most frequent 5 words in each topic group generated by plain LDA, DMR and nnLDA when setting the number of topic groups to be 4 in the synthetic dataset. The topic groups generated by plain LDA and DMR are very vague and it is very hard to distinguish which topic group is describing what combination of product and description, while the topic groups given by nnLDA are very distinguishable, i.e. topic group 1 is about (burger, quality), topic group 2 is about (TV, price), topic group 3 is about (TV, quality) and topic group

	plain LDA	DMR	nnLDA
Topic group 1	promising, rebate, sharp, increase, outstanding	pricey, unacceptable, juicy, pixilated	unreason, unacceptable, juicy delicious, nasty
Topic group 2	unreason, value, okay, steep, ecofriendly	ouch, steep, tasty, unreason, promotion	promotion, increase, tasty, economical, okay
Topic group 3	reason, accept, promotion, large, unacceptable	accept, fantastic, value reason, affordable	fresh, promising, fantastic, large, eco friendly
Topic group 4	fresh, reason, outstanding, ecofriendly, fantastic	sharp, delicious, accept, fresh, clear	reason, accept, value, steep, cheap

Table 3: Top words of groups generated by LDA, DMR and nnLDA

4 is about (burger, price). It identifies correctly the seed topics. Therefore, nnLDA outperforms plain LDA in grouping.

Additionally, based on the top words of topics generated by LDA, DMR and nnLDA, we are able to assign the most related category combination to a comment with respect to a model. Since we have the category combination of each comment, Table 4 shows the macro-recall, macro-precision and macro-F1 scores and micro-F1 of LDA, DMR and nnLDA, respectively, when training on the synthetic dataset, and the overall relative improvement of nnLDA. As the table shows, nnLDA outperforms plain LDA and DMR, which implies that nnLDA assigns more samples correctly to the right topic group. Therefore, in general, nnLDA improves the recall, precision and F1 scores.

	macro precision	macro recall	macro F1	micro F1
LDA	0.7238	0.7272	0.7211	0.7240
DMR	0.7238	0.7460	0.7313	0.7392
nnLDA	0.7401	0.7919	0.7536	0.7905
relative improvement from LDA	2.25%	8.90%	4.51%	9.19%
relative improvement from DMR	2.25%	6.15%	3.05%	6.94%

Table 4: Precision, recall and relative improvement of the synthetic dataset generated by LDA, DMR and nnLDA

In conclusion, nnLDA outperforms standard LDA and DMR in terms of the ability of topic grouping.

4.2.2 Perplexity

Figures 1 and 2 represent the log(perplexity) of plain LDA, DMR and nnLDA on the PTS and WIP datasets, respectively. Additionally, in Figure 2, for DMR and nnLDA, we not only conduct experiments on the dataset with the single feature (sector) as the side data, denoted as "DMR with single feature" and "nnLDA with single feature," but also on the dataset with two features (sector and channel) as side data, denoted as "DMR with two features" and "nnLDA with two features," respectively. The smallest log(perplexity) values generated by plain LDA and DMR are competitive to those of nnLDA for these two datasets. In Figure 1, the log(perplexity) value generated by plain LDA increases as the number of topic groups grows, while the log(perplexity) values generated by DMR and nnLDA decrease first and then increase as the number of topic groups increases on the PTS dataset. As it is shown in Figure 2, the log(perplexity) values generated by plain LDA and DMR increase as the number of topic groups grows on the WIP dataset. However, the log(perplexity) values generated by nnLDA decrease first and then increase as the number of topic groups increases on both of the aforementioned datasets. Moreover, we examine DMR and nnLDA models with two features on the WIP dataset, which take both sector and channel attributions as side data into account, in Figure 2. As we can observe, the minimum log(perplexity) generated by nnLDA with two features (sector and channel attributions) is better than that of nnLDA with the single feature (sector attribution), although the optimal number of topic groups occurs at a different point since more side data is provided. Consequently, plain LDA does not learn the datasets, and DMR is able to learn the small datasets. In contrast, nnLDA starts learning the datasets as the log(perplexity) value decreases in the beginning and finds an optimal number of topic groups, then it gets confused since the number of topic groups are more than needed. Furthermore, nnLDA with two features provides better log(perplexity) than nnLDA with the single feature. Therefore, nnLDA is more capable of understanding the datasets; both

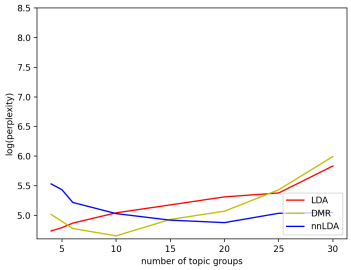

Fig. 1. PTS dataset

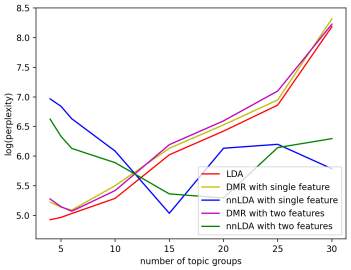

Fig. 2. WIP dataset

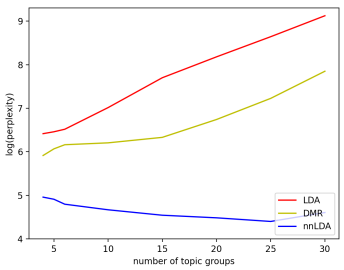

Fig. 3. DCL dataset

small and medium-size datasets with short comments.

When handling complex datasets, nnLDA's advantage becomes more pronounced. Figure 3 shows the log(perplexity) of plain LDA, DMR, and nnLDA on the DCL dataset. We observe that log(perplexity) for plain LDA and DMR increases with more topic groups, while nnLDA's log(perplexity) decreases initially before rising. nnLDA consistently yields lower log(perplexity) values, outperforming plain LDA and DMR in medium and large datasets with long comments while performing comparably on smaller datasets. However, as shown in Table 5, nnLDA requires slightly more training time—less than 10% slower than DMR—indicating a trade-off between accuracy and efficiency.

running time(s)	plain LDA	DMR	nnLDA
PTS	3	4	4
WIF	19	24	26
DCL	138	179	191

Table 5: Running time of different models on different datasets

In the following section, we study the classification problem of predicting the rating of each sample. In all the cases, we use 10-fold cross validation, which holds out 10% of the data for test purposes and trains the models on the remaining 90%. We apply nnLDA, plain LDA and DMR to find the probability of each sample to be assigned to each topic group and treat it as the feature matrix. Lastly, we train a classification model (xgboost (Chen and Guestrin, 2016)) on the feature matrix with the rating labels as the ground truth.

4.2.3 Classification

Figures 4, 5 and 6 depict the relative F1 scores of DMR and nnLDA with respect to plain LDA on the PTS, WIP, and DCL datasets, respectively. In Figure 4, the most distinguishable difference of F1 scores occurs when the number of topic groups is 15, where nnLDA has a gap of 0.032. In the meanwhile, DMR achieves its best performance at the same point with a gap of 0.030. Moreover, this chart shows that nnLDA outperforms plain LDA and DMR no matter what the number of topic groups is.

In Figure 5, when using the single feature (sector attribution), the biggest gaps of F1 scores happen when the number of topic groups is 15 for DMR and 25 for nnLDA. The biggest gap between nnLDA and plain LDA is 0.016, while the largest gap between DMR and plain LDA is 0.003. Considering models using two features (sector and channel attributions) as the side data, the highest relative F1 score given by nnLDA with two features is 0.022 with 15 topic groups, compared with 0.004 produced by DMR with 10 topic groups. Although plain LDA provides a slightly higher F1 score than nnLDA when applying 5 topic groups, nnLDA outperforms plain LDA and DMR significantly given any other number of topic groups. In Figure 6, the highest relative F1 score given by nnLDA is 0.022 with 25 topic groups, compared with 0.003 given by DMR for 6 topic groups. Moreover, this figure shows that nnLDA outperforms plain LDA whatever the number of topic groups is.

Therefore, nnLDA performs better than plain LDA and DMR when predicting the rating given customer's comments and product information in all datasets.

4.2.4 Comment Generation

In this section, we compare the comments generated by nnLDA with plain LDA and DMR. We set the number of topic groups to be 5 since all of plain LDA, DMR and nnLDA have relatively low perplexity scores based on Figures 1 and 2, and comparable F1 scores based on Figures 4 and

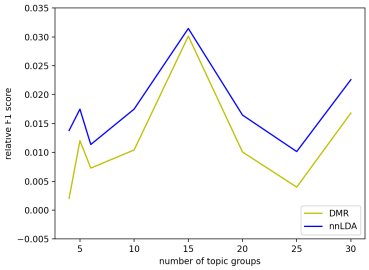

Fig. 4. PTS

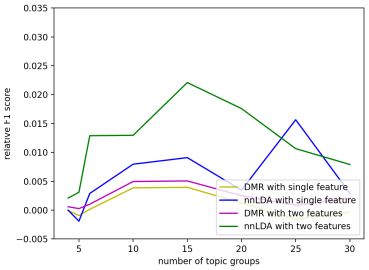

Fig. 5. WIP

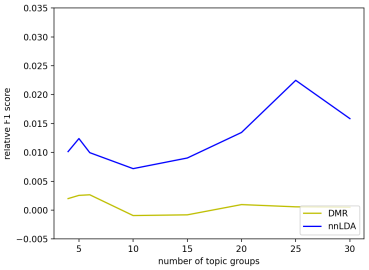

Fig. 6. DCL

5 on the PTS and WIP datasets. A comment is generated based on the topic-document probability of the sample and the topic-word distribution. More precisely, for DMR and LDA, the prior α is generated based on the side data (sector) first while α is fixed in plain LDA. Next, a comment is created by selecting the top words which have the highest score computed by adding the products of the topic-document probability and topic-word for each word. Then, we randomly pick 50 comments that contain a certain level of information, for example, we rule out comments like "N/A." Meanwhile, in order to evaluate the quality of comment generation, we employed 50 PhD students. Each one of them assessed a pair of comments (one based on plain LDA or DMR, and the other one based on nnLDA) for the same side data and they provided an assessment as to which one is better.

	Number of generated comments	
	PTS	WIP
plain LDA < nnLDA	15	22
plain LDA > nnLDA	11	9
plain LDA $\sim$ nnLDA	24	19
DMR < nnLDA	16	20
DMR > nnLDA	11	10
DMR $\sim$ nnLDA	23	20

Table 6: Comparison of the generated comments on different datasets

The upper left three values in Table 6 show the comparison of the generated comments given by plain LDA and nnLDA on the PTS dataset. Based on the table, among all these 50 samples, nnLDA generates more accurate comments in 15 samples, while plain LDA does better in 11 samples, and the two are tied for the remaining 24 samples. The lower left three values in Table 6 show the comparison of the generated comments given by DMR and nnLDA on the PTS dataset. Based on the table, among all these 50 samples, nnLDA generates more accurate comments in 16 samples, while DMR does better in 11 samples, and the two are tied for the remaining 23 samples. On the PTS dataset, nnLDA generates in $\frac{15-11}{50} = 8\%$ more reasonable comments compared to plain LDA, and in $\frac{16-11}{50} = 10\%$ more comparing to DMR.

The right column in Tables 6 shows the comparison of the generated comments given by plain LDA and nnLDA, and DMR and nnLDA on the WIP dataset, respectively. The observations and conclusions are similar. Furthermore, the advantage in number is more obvious on the WIP dataset, i.e. the improvement of nnLDA compared to plain LDA is as large as $\frac{22-9}{50} = 26\%$ and the improvement from DMR to nnLDA is $\frac{20-10}{50} = 20\%$. Therefore, taking generated comments into consideration, nnLDA generates more reasonable comments than plain LDA and DMR for both small and medium-sized datasets.

5 Conclusion

Our experiments confirm that integrating side data via a neural network into the LDA framework can significantly improve performance on multiple tasks. In particular, nnLDA consistently achieves higher log-likelihoods, and its adaptive prior—learned directly from side data—leads to better topic grouping, lower perplexity, and enhanced classification and comment generation. The theoretical guarantees (see Appendix A) further support these empirical findings.

Future work will explore alternative neural network architectures to better adapt to various types of side data and will extend the evaluation to a broader range of datasets. Overall, nnLDA provides a comprehensive framework for integrating auxiliary information into topic modeling, thereby offering significant improvements over existing approaches.

Limitations

The nnLDA assumes that side data is relevant and beneficial for the topic modeling process. However, in real-world applications, side data may sparse, noisy, or not correlate with the textual content. In such cases, the model could produce misleading or less coherent topic structures, reducing its effectiveness. Future work could explore adaptive models that can adjust their reliance on side data based on its relevance.

Although nnLDA shows improvements over traditional models like LDA and DMR, it has not been compared to more recent advances in topic modeling, such as transformer-based models (e.g., BERT-LDA) or other deep generative models. These models may offer additional benefits such as better semantic coherence or reduced reliance on side data, suggesting that further benchmarks are needed.

Ethics Statement

This work was conducted using a combination of publicly available synthetic data and proprietary datasets that have been anonymized and aggregated to protect individual privacy. Our research focuses solely on improving topic modeling techniques and does not involve any collection or analysis of personally identifiable information. All experiments were performed in accordance with applicable ethical guidelines and institutional policies, ensuring that no harm or bias is introduced in the processing and analysis of data. We believe that the methodologies and findings presented in this paper adhere to ethical research practices and contribute positively to the development of transparent and accountable machine learning models.

References

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. 2008. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 9:1981–2014.

Adrian Benton, Michael J. Paul, Braden Hancock, and Mark Dredze. 2016. Collective supervision of topic models for predicting surveys with social media. In *AAAI*.

David M. Blei and Michael I. Jordan. 2003. Modeling annotated data. In *SIGIR*.

David M. Blei and Jon D. McAuliffe. 2007. Supervised topic models. In *NIPS*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of machine learning research*, 3:993–1022.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD*.

Laura Dietz, Steffen Bickel, and Tobias Scheffer. 2007. Unsupervised prediction of citation influences. In *ICML*.

Elena A. Erosheva, Stephen E. Fienberg, and John Lafferty. 2004. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5220 – 5227.

P. Haritha and P. Shanmugavadivu. 2024. Optimized latent-dirichlet-allocation based topic modeling–an empirical study. In *Speech and Language Technologies for Low-Resource Languages*, pages 412–419. Springer Nature Switzerland.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*.

Di Jiang, Kenneth Wai-Ting Leung, Wilfred Ng, and Hao Li. 2013. Beyond click graph: Topic modeling for search engine query log analysis. In *ICDSAA*.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. 2008. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*.

Fei-Fei Li and Pietro Perona. 2005. A Bayesian hierarchical model for learning natural scene categories. In *ICCV*.

Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. 2007. Topic and role discovery in social networks with experiments on Enron and academic email. *Journal of artificial intelligence research*, 30:249–272.

David Mimno and Andrew McCallum. 2008. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *UAI*.

Jonas Rieger, Carsten Jentsch, and Jörg Rahnenführer. 2021. RollingLDA: An update algorithm of Latent Dirichlet Allocation to construct consistent time series from textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. *ArXiv*, abs/1207.4169.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. 2005. Learning hierarchical models of scenes, objects, and parts. In *ICCV*.

Chong Wang, David M. Blei, and Fei-Fei Li. 2009. Simultaneous image classification and annotation. In *ICCV*.

Xingyu Wang, Lida Zhang, and Diego Klabjan. 2020. Keyword-based topic modeling and keyword selection. *ArXiv*, abs/2001.07866.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. *Proceedings of the 12th ACM SIGKDD*.

Xuerui Wang, Natasha Mohanty, and Andrew McCallum. 2005. Group and topic discovery from relations and their attributes. In *NIPS*.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. 2019. Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity. In *NIPS*.

Jun Zhu, Amr Ahmed, and Eric P. Xing. 2012. MedLDA: maximum margin supervised topic models. *Journal of machine learning research*, 13:2237–2278.

Appendix

A. Analytical comparison of standard LDA and nnLDA

Note that a Dirichlet random vector $\theta = (\theta_1, \theta_2, \cdots, \theta_K)$ has the following probability density:

$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1},$$

where K is the number of topic groups, α is the prior of the Dirichlet distribution and θ takes values in the $(K-1)$-simplex. Then, the generative process implies that the conditional distribution of the nnLDA model of a document $d = (\mathbf{w}, \mathbf{s})$ is

$$\begin{aligned}
&P_1(\mathbf{w} \mid \mu, \sigma, \gamma, \beta) = \tilde{\tilde{P}}_1(\mathbf{w} \mid \mathbf{s}, \gamma, \beta) \\
&= \int \tilde{\tilde{p}}(\theta \mid \mathbf{s}, \gamma) \left(\prod_{n=1}^{N} \sum_{z_k} \tilde{p}(z_k \mid \theta) \tilde{p}(w_n \mid z_k, \beta) \right) \mathrm{d}\theta \\
&= \int \tilde{p}(\theta \mid \mu, \sigma, \gamma) \left(\prod_{n=1}^{N} \sum_{z_k} \tilde{p}(z_k \mid \theta) \tilde{p}(w_n \mid z_k, \beta) \right) \mathrm{d}\theta \\
&= \int \tilde{p}(\theta \mid \mu, \sigma, \gamma) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) \mathrm{d}\theta,
\end{aligned}$$

which in turn yields

$$\begin{aligned}
&P_1(D \mid \mu, \sigma, \gamma, \beta) \\
&= \mathbb{E}\left[\int \tilde{p}(\theta_d \mid \mu, \sigma, \gamma) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \tilde{p}(z_{d_k} \mid \theta_d) \tilde{p}(w_{d_n} \mid z_{d_k}, \beta) \right) \mathrm{d}\theta_d \right] \\
&= \mathbb{E}\left[\int \tilde{p}(\theta_d \mid \mu, \sigma, \gamma) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) \mathrm{d}\theta_d \right].
\end{aligned}$$

where $\tilde{p}(\theta_d \mid \mu, \sigma, \gamma) = \tilde{\tilde{p}}(\theta_d \mid \mathbf{s}, \gamma) = p(\theta_d \mid g(\gamma; \mathbf{s})) = p(\theta_d \mid \alpha_d)$ for a corpus D.

The nnLDA model represented above is a probabilistic graphical model with three levels. Parameters μ, σ, γ and β are corpus-level parameters, which are assumed to be sampled once in the generative process of a corpus. Variables α_d and θ_d are document-level variables, which are sampled once per document. Finally, w_{d_n} and z_{d_k} are word-level variables, sampled once for each word in each document. In the rest of this section, we provide an analytical comparison of standard LDA and nnLDA.

Compared to standard LDA, nnLDA employs an extra neural network g to generate document-level variable α_d. Since nnLDA is "richer" than LDA, we expect that it should produce a higher likelihood. Without assumptions on $g(\gamma; \cdot)$ this does not hold since, for example, $g(\gamma; \cdot)$ can map everything to a constant vector different from the prior used by LDA. As a result, in order for the statement to hold the network must be expressive. The question to consider is whether a neural network is capable of memorizing arbitrary side data of a given size. We tackle this question by introducing the concept of finite sample expressivity which is an extension of a similar definition in (Yun et al., 2019). Given the definition, if $g(\gamma; \cdot)$ has finite sample expressivity, nnLDA at least can find the optimal α^* used in standard LDA.

Definition 1. *Function $g(\gamma; \cdot)$ has finite sample expressivity if for all inputs $x_i \in \mathrm{R}^{d_x}, 1 \leq i \leq N$ and for all $y_i \in [-M, +M]^{d_y}, 1 \leq i \leq N$ for some constant $M > 0$, there exists a parameter γ such that $g(\gamma; x_i) = y_i$ for every $1 \leq i \leq N$.*

Based on Definition 1, Theorem 3.1 shown in (Yun et al., 2019) provides a specific set of constraints, i.e. any 3-layer (i.e., 2-hidden-layer) ReLU FCNN with hidden layer widths d_1 and d_2 can fit any arbitrary dataset if $d_1 d_2 \geq 4 N d_y$, where d_y and N are the dimension of the label and the number of samples, respectively. By extending the aforementioned theorem, Proposition 3.4 and Theorem 4.1 in

(Yun et al., 2019) argue that any FCNN given constraints on the number of neurons in each layer is able to have finite sample expressivity. In the following, we assume that $g(\gamma; \cdot)$ has finite sample expressivity. Therefore, given K and any α^* representing the number of topic groups and optimal parameters in LDA, since $\alpha^* \in [-M, +M]^K$ for some constant M, there exists a γ_1 such that, for all inputs $\mathbf{s}_i$ and α^*, $g(\gamma_1; \mathbf{s}_i) = \alpha^*$ for all $1 \leq i \leq N$.

We next prove that the optimized probability of nnLDA is at least as good as that of plain LDA. Let α^* and β^* be optimal solutions to $P_2 = \max_{\alpha,\beta} P(D|\, \alpha, \beta)$ of LDA, meanwhile, let μ^*, σ^* and γ^* be optimal solutions to $P_1 = \max_{\mu,\sigma,\gamma} P_1(D|\, \mu, \sigma, \gamma, \beta^*)$ of nnLDA (see Appendix B for formal definitions).

Theorem 1. *If α^*, β^* are optimal solutions to LDA, then there exists optimal solutions μ^*, σ^* and γ^* to nnLDA such that*

$$P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) \geq P_2(D \mid \alpha^*, \beta^*).$$

Proof. See Appendix B. □

While Theorem 1 asserts that when it comes to model fit nnLDA fits the data better than LDA, it does not provide a gap statement. If the side data provides positive influence during the learning process by a constant C, then, due to the independence of words, topics and documents, we are able to argue that the optimized probability is at least improved by $C - 1$.

Theorem 2. *For any document $(\boldsymbol{w}, \boldsymbol{s}) \in (D, S)$, if $\hat{p}(w_i \mid \alpha^*, \beta^*) \neq 0$ for all i, and there exists a positive constant $C > 1$ such that $\prod_{i=1}^{N} \tilde{p}(w_i \mid \gamma^*, \beta^*, \mu^*, \sigma^*) \geq C \prod_{i=1}^{N} \hat{p}(w_i \mid \alpha^*, \beta^*)$ for every $w_i \in \boldsymbol{w}$, and if D in P_1 and D in P_2 follow the same distribution, then*

$$\frac{P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) - P_2(D \mid \alpha^*, \beta^*)}{P_2(D \mid \alpha^*, \beta^*)} \geq C - 1.$$

Proof. See Appendix D for a formal proof. □

The assumption on $\hat{p}(w_i \mid \alpha^*, \beta^*)$ in Theorem 2 is reasonable since it indicates that all documents are not randomly generated. The positive constant C in the assumption captures the improvement given by the side data. In other words, as long as the side data has positive impact on the text data, this assumption holds. Next, we link the existence of C to lift from data mining. Let us define lift as

$$l(d) = \frac{P(\mathbf{w})P(\mathbf{s})}{P(\mathbf{w}, \mathbf{s})}$$

with $d = (\mathbf{w}, \mathbf{s})$. Lift measures the dependency level of words $\mathbf{w}$ and side data $\mathbf{s}$. If $l(d) < 1$ for d with N words and $P(\mathbf{s}) > 0$, we have

$$P(\mathbf{s}) \prod_{n=1}^{N} P(w_n) = P(\mathbf{w})P(\mathbf{s}) < P(\mathbf{w}, \mathbf{s}) = P(\mathbf{s}) \prod_{n=1}^{N} P(w_n|\mathbf{s}),$$

and in turn

$$\prod_{n=1}^{N} P(w_n) < \prod_{n=1}^{N} P(w_n|\mathbf{s}),$$

and

$$\prod_{n=1}^{N} \hat{p}(w_n \mid \alpha^*, \beta^*) < \prod_{n=1}^{N} \tilde{p}(w_n \mid \gamma^*, \beta^*, \mu^*, \sigma^*).$$

This implies that there exists $C > 1$. In summary, when $l(d) < 1$ and $P(\mathbf{s}) > 0$ for each d in the corpus, Theorem 2 holds. Lift essentially measures the dependency of $\mathbf{w}$ and $\mathbf{s}$, which is widely used in data mining. The condition indicates that the side data helps to link the words to the documents they are

more likely to be in. Informally, in the proof, due to the independence assumption of words, topics and documents in nnLDA, the generative probability of nnLDA for a corpus can be reformulated as a product of $\tilde{p}(\theta_d \mid \mu^*, \sigma^*, \gamma^*)$ and conditional probability of words $\tilde{p}(w_n \mid \theta_d, \beta^*)$. Likewise, the same property holds for plain LDA. Lastly, given a relationship between documents $d = \mathbf{w}$ and $d = (\mathbf{w}, \mathbf{s})$ as an expression of the conditional probability of words, we are able to build a connection of the optimized probabilities between nnLDA and LDA.

B. Probability Distribution of LDA

Given the generative process of LDA, which is formally presented in (Blei et al., 2003), we obtain the marginal distribution of a document $d = \mathbf{w}$ with text only as

$$P_2(\mathbf{w} \mid \alpha, \beta) = \int \hat{p}(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_k} \hat{p}(z_k \mid \theta) \hat{p}(w_n \mid z_k, \beta) \right) \mathrm{d}\theta$$

$$= \int \hat{p}(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) \mathrm{d}\theta,$$

which in turn yields

$$P_2(D \mid \alpha, \beta) = \mathbb{E}\left[\int \hat{p}(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \hat{p}(z_{d_k} \mid \theta_d) \hat{p}(w_{d_n} \mid z_{d_k}, \beta) \right) \mathrm{d}\theta_d \right]$$

$$= \mathbb{E}\left[\int \hat{p}(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) \mathrm{d}\theta_d \right],$$

where $\hat{p}(\theta_d \mid \alpha) = p(\theta_d \mid \alpha)$.

C. Proof of Theorem 1

Proof. By finite sample expressivity of $g(\gamma; \cdot)$, there exists a model with parameters γ_1 such that

$$g(\gamma_1; \mathbf{s}) = \alpha^*,$$

which in turn yields

$$\tilde{\hat{p}}(\theta \mid \mathbf{s}, \gamma_1) = \hat{p}(\theta \mid g(\gamma_1; \mathbf{s})) = \hat{p}(\theta \mid \alpha^*).$$

Therefore,

$$P_2(D \mid \alpha^*, \beta^*) = \tilde{\hat{P}}_1(D \mid S, \gamma_1, \beta^*) = P_1(\mu^*, \sigma^*, \gamma_1, \beta^*).$$

Since nnLDA also optimizes over the network parameter γ, we have

$$P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) \geq P_1(D \mid \mu^*, \sigma^*, \gamma_1, \beta^*),$$

and thus,

$$P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) \geq P_2(D \mid \alpha^*, \beta^*).$$

□

D. Proof of Theorem 2

Proof. Note that

$$\begin{aligned}
&\frac{P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) - P_2(D \mid \alpha^*, \beta^*)}{P_2(D \mid \alpha^*, \beta^*)} \\
=&\frac{\mathbb{E}\left[\int \tilde{p}(\theta_d \mid \mu^*, \sigma^*, \gamma^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \tilde{p}(z_{d_k} \mid \theta_d) \tilde{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \mathrm{d}\theta_d\right]}{\mathbb{E}\left[\int \hat{p}(\theta_d \mid \alpha^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \hat{p}(z_{d_k} \mid \theta_d) \hat{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \mathrm{d}\theta_d\right]} \\
&- \frac{\mathbb{E}\left[\int \hat{p}(\theta_d \mid \alpha^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \hat{p}(z_{d_k} \mid \theta_d) \hat{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \mathrm{d}\theta_d\right]}{\mathbb{E}\left[\int \hat{p}(\theta_d \mid \alpha^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \hat{p}(z_{d_k} \mid \theta_d) \hat{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \mathrm{d}\theta_d\right]}. \qquad (2)
\end{aligned}$$

Since

$$\begin{aligned}
&\tilde{p}(\theta_d \mid \mu^*, \sigma^*, \gamma^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \tilde{p}(z_{d_k} \mid \theta_d) \tilde{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \\
=&\tilde{p}(\theta_d \mid \mu^*, \sigma^*, \gamma^*) \left(\prod_{n=1}^{N} \tilde{p}(w_{d_n} \mid \theta_d, \beta^*)\right) = \prod_{n=1}^{N} \tilde{p}(w_{d_n} \mid \gamma^*, \beta^*, \mu^*, \sigma^*)
\end{aligned}$$

and

$$\begin{aligned}
&\hat{p}(\theta_d \mid \alpha^*) \left(\prod_{n=1}^{N} \sum_{z_{d_k}} \hat{p}(z_{d_k} \mid \theta_d) \hat{p}(w_{d_n} \mid z_{d_k}, \beta^*)\right) \\
=&\hat{p}(\theta_d \mid \alpha^*) \left(\prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \theta_d, \beta^*)\right) = \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*),
\end{aligned}$$

equation (2) could be further simplified as

$$\begin{aligned}
&\frac{P_1(D \mid \mu^*, \sigma^*, \gamma^*, \beta^*) - P_2(D \mid \alpha^*, \beta^*)}{P_2(D \mid \alpha^*, \beta^*)} \\
=&\frac{\mathbb{E}\left[\int \prod_{n=1}^{N} \tilde{p}(w_{d_n} \mid \mu^*, \sigma^*, \gamma^*, \beta^*) \mathrm{d}\theta_d\right] - \mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]}{\mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]} \\
\geq&\frac{\mathbb{E}\left[\int C \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right] - \mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]}{\mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]} \\
=&\frac{C \cdot \mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right] - \mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]}{\mathbb{E}\left[\int \prod_{n=1}^{N} \hat{p}(w_{d_n} \mid \alpha^*, \beta^*) \mathrm{d}\theta_d\right]} = C - 1.
\end{aligned}$$

□