Intelligent Data Analysis 28 (2024) 161–183
DOI 10.3233/IDA-227202
IOS Press

Supervised probabilistic latent semantic analysis with applications to controversy analysis of legislative bills

Eyor Alemayehu* and Yi Fang
Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA

Abstract. Probabilistic Latent Semantic Analysis (PLSA) is a fundamental text analysis technique that models each word in a document as a sample from a mixture of topics. PLSA is the precursor of probabilistic topic models including Latent Dirichlet Allocation (LDA). PLSA, LDA and their numerous extensions have been successfully applied to many text mining and retrieval tasks. One important extension of LDA is supervised LDA (sLDA), which distinguishes itself from most topic models in that it is supervised. However, to the best of our knowledge, no prior work extends PLSA in a similar manner sLDA extends LDA by jointly modeling the contents and the responses of documents. In this paper, we propose supervised PLSA (sPLSA) which can efficiently infer latent topics and their factorized response values from the contents and the responses of documents. The major challenge lies in estimating a document's topic distribution which is a constrained probability that is dictated by both the content and the response of the document. To tackle this challenge, we introduce an auxiliary variable to transform the constrained optimization problem to an unconstrained optimization problem. This allows us to derive an efficient Expectation and Maximization (EM) algorithm for parameter estimation. Compared to sLDA, sPLSA converges much faster and requires less hyperparameter tuning, while performing similarly on topic modeling and better in response factorization. This makes sPLSA an appealing choice for latent response analysis such as ranking latent topics by their factorized response values. We apply the proposed sPLSA model to analyze the controversy of bills from the United States Congress. We demonstrate the effectiveness of our model by identifying contentious legislative issues.

Keywords: Probabilistic latent semantic analysis, topic models, controversy analysis

1. Introduction

Hofmann [1] introduced Probabilistic Latent Semantic Analysis (PLSA), which is also known as Probabilistic Latent Semantic Indexing (PLSI) when used in information retrieval and text mining [2]. The basic idea of PLSA is to treat the words in each document as observations from a mixture model where the components of the model are word distributions for latent topics. The selection of the latent topics is controlled by a set of mixing weights such that words in the same document share the same mixing weights. PLSA was initially proposed for text-based applications that do indexing, retrieval, mining, and clustering. Later, its use was expanded to other fields including collaborative filtering [3], computer vision [4], and audio processing [5].

PLSA can be viewed as a probabilistic version of the seminal work on latent semantic analysis [6], which revealed the utility of the singular value decomposition of the document-term matrix. PLSA is

*Corresponding author: Eyor Alemayehu, Department of Computer Science and Engineering, Santa Clara University, 500 El Camino Real, Santa Clara, CA 95053, USA. E-mail: ealemayehu@scu.edu.

the precursor of probabilistic topic models which are widely used nowadays including Latent Dirichlet Allocation (LDA) [7]. The basic generative processes of PLSA and LDA are very similar. In PLSA, the topic mixture is conditioned on each document, while the topic mixture in LDA is drawn from a conjugate Dirichlet prior. Theoretically, PLSA is equivalent to MAP estimated LDA under a uniform prior [8]. The PLSA model does not make any assumptions about how the mixture weights are generated and thus its generative semantics are not well defined [7]. Consequently, there is no natural way to predict a previously unseen document. On the other hand, the LDA model is more complex and cannot be solved by exact inference. Gibbs sampling [9] and variational inference [7] are often used for inference in LDA type of topic models. However, these methods scale poorly to large datasets. Variational inference requires dozens of expensive passes over the entire dataset, and Gibbs sampling requires multiple Markov chains [10]. In contrast, the parameter estimation and inference of PLSA can be efficiently done by the EM algorithm.

PLSA and LDA are the two most representative topic models. Various empirical comparisons have been conducted between them. Blei et al. [7] shows that LDA outperforms PLSA in the perplexity of new documents. On the other hand, Lu et al. [11] conduct a systematic empirical study of PLSA and LDA on three representative IR tasks, including document clustering, text categorization, and ad-hoc retrieval. They found that LDA and PLSA tend to perform similarly on these tasks. Furthermore, the performance of LDA on all tasks is quite sensitive to the setting of its hyperparameters, and the optimal setting of hyperparameters varies according to how the model is used in a task.

The original PLSA and LDA models as well as most of their variants are unsupervised models. Many real-world text documents are associated with a response variable connected to each document such as the number of stars given to a movie, the number of times a news article was downloaded, or the category of a document. Incorporating such information into latent aspect modeling could guide a topic model towards discovering semantically more salient statistical patterns that may be more interesting or relevant to the user's task. Thus, a very important extension of LDA is supervised LDA (sLDA) [12]. sLDA jointly models the content and responses of documents in order to find latent topics that best predict the responses of documents.

In this paper, we propose supervised Probabilistic Latent Semantic Analysis (sPLSA) by extending PLSA to learn from the responses of documents. For PLSA, our proposed model is the analog of what sLDA is to LDA. The major challenge lies in estimating a document's topic distribution which is a constrained probability that is dictated by both the content and the response of the document. We introduce an auxiliary variable to transform the constrained optimization problem to an unconstrained optimization problem. This allows us to derive an efficient EM algorithm to estimate the parameters of our model. Compared to sLDA, sPLSA is much more efficient and requires less hyperparameter tuning, while performing similarly on topic modeling and better in response factorization. This makes sPLSA the ideal choice for latent response analysis such as ranking latent topics by their factorized response values. We utilize the sPLSA model to analyze the controversy of bills from the United States Congress. We demonstrate the effectiveness of our model by identifying contentious legislative issues. The contributions of the paper can be summarized as follows.

- We propose a novel supervised PLSA model which can efficiently infer latent topics and their factorized response values from the contents and the responses of documents.
- We derive an efficient EM algorithm to estimate the parameters of the model.
- We utilize sPLSA and sLDA to analyze the controversy of bills from the United States Congress. We demonstrate the effectiveness of sPLSA over sLDA as part of this analysis.

2. Related work

2.1. Probabilistic topic models

In 1999, three papers [1,2,13] introduced the model of Probabilistic Latent Semantic Analysis. One variant of the model appeared in 1998 [14] and all these models were originally discussed in an earlier technical report [15]. PLSA was a probabilistic implementation of latent semantic analysis (LSA) introduced by Deerwester et al. [6]. LSA was extended from the vector space model. It aimed to represent documents in a low dimensional vector space consisting of common semantic factors. Differing from LSA in projecting document or word vectors into the latent semantic space, PLSA extracted the aspects related to documents. This aspect model was interpreted as a mixture model containing latent semantic mixtures. Parameters of mixture probabilities were estimated by the maximum-likelihood (ML) principle. PLSA did not provide a straightforward way to make inferences about new documents not seen in the training data and the parameterization of the model was susceptible to overfitting. Latent Dirichlet Allocation (LDA) addressed these limitations by proposing a Bayesian probabilistic topic model.

PLSA and LDA established the field of probabilistic topic models. Many extensions of the two basic models have been proposed. In Zhai et al. [16], PLSA was extended to include a background component to explain the non-informative background words and a cross-collection mixture model was proposed to support comparative text mining. Mei and Zhai [17] propose a general contextual text mining model which is an extension of PLSA to incorporate context information. They further regularize PLSA with a harmonic regularizer based on a graph structure in the data [18]. One active area of topic modeling research is how to relax and extend the assumptions of PLSA and LDA to uncover more sophisticated structure in the texts. For example, the work by Rosen-Zvi et al. [19] extends LDA to include authorship information. Recently, probabilistic topic models are proposed for unsupervised many-to-many object matching [20] and cross-lingual tasks [21]. There are many other topic models proposed. Blei [22] gives an overview of the field of probabilistic topic models.

The original PLSA and LDA and most of their variants are unsupervised models. Blei and McAuliffe [12] proposed supervised LDA (sLDA) to capture real-valued document rating as a regression response. The generative process of sLDA is similar to LDA, but with an additional step: draw a reponse variable. The sLDA model is trained by maximizing the joint likelihood of the contents and the responses of documents. They tested sLDA on two real-world datasets: movie reviews with ratings and web pages with popularity, and the experimental results demonstrated the advantages of sLDA versus regularized regression, and versus an unsupervised LDA analysis followed by a separate regression. Other extensions include multi-class sLDA [23], which directly captures discrete labels of documents as a classification response; and discriminative LDA (DiscLDA) [24], which also performs classification, but with a mechanism different from that of sLDA; and MedLDA [25], which leverages the maximum margin principle for estimation of latent topical representations. Recently, Jameel et al. [26] integrate class label information and word order structure into a supervised topic model for document classification. More variants of supervised topic models can be found in a number of applied domains, such as Labeled LDA [27], automatic summarization of changes in dynamic text collections [28], modeling of numerical time series [29], inferring topic hierarchies [30], and query expansion [31]. In computer vision, several supervised topic models have been designed for understanding complex scene images [32,33]. Mimno and McCallum [34] also proposed a topic model for considering document-level meta-data; for example, publication date and venue of a paper.

Most of the above supervised topic models are based on LDA. There exist very few work on extending PLSA to the supervised setting. One such work was to use the spoken content of a multimedia document

as a query for retrieving similar or relevant documents [35]. The query was used to train the model in a supervised fashion with respect to a query-document similarity objective function. Fergus et al. [36] extend PLSA to include spatial information in a translation and scale invariant manner, and utilized this modified PLSA model to learn an object category. Another work added a category-topic distribution in PLSA for human action recognition [37]. However, these models do not associate the topic distribution of the document with the response variable. Consequently, the discovered topics may not be indicative of the response. Aliyanto et al. [38] proposed a version of supervised PLSA to estimating technology readiness level, but they assumed the topics of each word in a document are observed which are actually not available in many real-world applications. In this paper, we follow the way LDA was extended to sLDA by directly associating the documents' topic distributions with the response. The response is at the document level instead of the word level and it is more readily accessible. The learned topics depend on both the document's content and response. To the best of our knowledge, no prior work has extended PLSA in a similar manner.

Recently, with the rise of deep learning, novel topic models based on neural networks have been proposed. Salakhutdinov and Hinton [39] proposed a two layer restricted Boltzmann machine (RBM) called the replicated-softmax to extract low level latent topics from a large collection of unstructured documents. Larochelle and Lauly [40] proposed a neural auto-regressive topic model inspired from the replicated softmax model but replacing the RBM model with a neural auto-regressive distribution estimator (NADE). Kingma and Welling [41] proposed variational autoencoders by combining topic modeling and neural networks. Cao et al. [42] proposed neural topic model (NTM), and it is supervised extension (sNTM) where words and documents embedding are combined. Moody [43] proposed the lda2vec, a model combining LDA and word embeddings. Dieng et al. [44] integrated to a recurrent neural network based language model global word semantic information extracted using a probabilistic topic model. Gupta et al. [45] integrated to an LSTM recurrent neural network, a neural auto-regressive topic model. Murakami and Chakraborty [46] investigated the use of word embedding with NTM for interpretable topics from short texts. Grootendorst [47] proposed BERTopic to generate document embedding with pre-trained transformer-based language models and then produce topic representations with the class-based TF-IDF procedure. Two recent surveys [48,**?**] provided comprehensive reviews of neural topic models, with nearly a hundred models developed and a wide range of applications in neural language understanding such as text generation, summarization and language models. Despite the popularity of deep learning, our work has focused on traditional probabilistic methods because they are often easier to implement and more efficient to train, which may be more suitable in resource constrained environments where only limited computation and storage are accessible. Nevertheless we will explore to combine the proposed model with neural networks in a future work.

2.2. Controversy analysis of legislative bills

Legislative voting is a major area of research. Most of the research is focused on the ideal point estimation of the ideological positions of legislators. This is primarily for the purpose of predicting their voting patterns. An early work in this area presented a spatial model of legislative voting [50]. Londregan [51] estimated the preferred positions of legislators by modeling the legislative agenda. Cox and Poole [52] used a spatial model to assess the role of partisanship in influencing the votes of legislators. Variational methods were applied to predict votes [53]. Thomas et al. [54] modeled voting behavior from congressional debate transcripts. Gerrish and Blei [55] demonstrated roll call predictive models which link legislative text with legislative sentiment. They [56] further derived approximate posterior inference

Table 1
Notations

$\mathcal{D}$	Corpus of documents	d	A document in $\mathcal{D}$
w	A word that occurs in $\mathcal{D}$	k	A topic
$n(d,w)$	Count of word w in d	K	Total number of topics
N	Total number of words	N_d	Number of words in d
M	Total number of documents	θ_{dk}	$P(k\|d)$
$\boldsymbol{\theta_d}$	Topic distribution of d	$\boldsymbol{\Theta}$	Matrix of all θ_{dk}
β_{kw}	$P(w\|k)$	$\boldsymbol{\beta_k}$	Word distribution of k
$\boldsymbol{\beta}$	Matrix of all β_{kw}	Z_{dn}	Topic of the n^{th} word in d
W_{dn}	n^{th} word in document d	$\boldsymbol{W}$	Matrix of all W_{dn}
c_d	Response of d	$\boldsymbol{c}$	Vector of all c_d
v_k	Regression coefficient on topic k	$\boldsymbol{v}$	Vector of all v_k
σ^2	Variance of the Gaussian noise		

algorithms based on variational methods to predict the positions of legislators. Fang et al. [57] analyzed public statements from legislators to build a contrastive opinion model of the legislators. Gu et al. [58] conducted ideal point estimations of legislators on the latent topics of voted documents.

Some of the work cited above utilized topic models. For example, Gerrish and Blei [55] extended LDA to build a generative model of votes and bills called the ideal point topic model. The model infers two bill related latent variables. One of the latent variables explains bills that all legislators will vote for or against while the other variable explains bills that do not have unanimous approval or disapproval. In addition, the model infers a latent variable for the legislators' ideal points. Another example, Fang et al. [57] present the cross-perspective topic model which unifies two identically extended LDA models to contrast the opinion words of a bipolar legislative body. The opinion words reflect the subjective positions of the polar entities on various topics. The model discriminates between opinion words and topics words by treating them as two separate observed variables.

On the broader field of controversy analysis, much work has been done detecting contradictions in textual data. One of the early works studied the dynamics of conflicting opinions in texts by visually inspecting graphs [59]. Tsytsarau et al. [60] further investigated two types of contradictions, namely, "overlapping contradicting opinions" and "change of sentiment". Many supervised learning approaches have been proposed for classifying texts into one of the two opposing opinions using annotated controversial corpora including sentences [61], documents [62] and document collections [61]. Some recent work addresses the task of identifying controversial contents on Wikipedia [63,64,65] and on social media [66, 67,68].

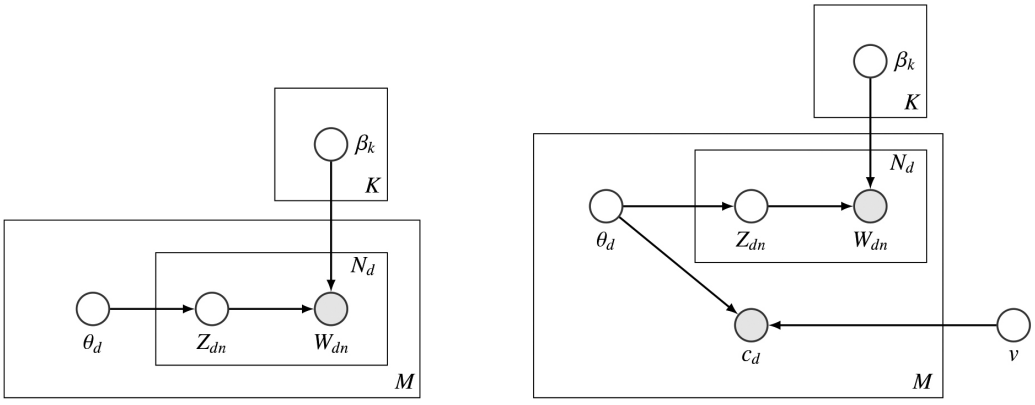
3. Supervised PLSA

3.1. Notations

Assume the corpus $\mathcal{D}$ contains M documents with K topics. N_d is the number of words in document d. Each document d has two set of observed variables: W_{dn}, which is the n^{th} word of d; and c_d, which is the response of d, such as the rating of a review. Table 1 includes the main notations in the paper.

3.2. Generative process

Similar to many other topic models, sPLSA assumes that a document consists of multiple topics. Therefore, there is a distribution $\theta_{\mathbf{d}}$ over a fixed number of K topics for each document d. Like PLSA,



(a) PLSA (b) sPLSA

Fig. 1. Graphical model representation of (a) PLSA and (b) sPLSA.

this distribution is a multinomial distribution where each element θ_{dk} in the vector represents the probability that topic k appears in document d, i.e., $\theta_{dk} = P(k|d)$. In addition, we assume each topic represents a distribution over words w in a fixed vocabulary of size V, denoted by $\beta_{\mathbf{k}}$. This distribution is also a multinomial distribution where each element β_{kw} represents the probability that term w is chosen by topic k, i.e., $\beta_{kw} = P(w|k)$.

The essential difference between PLSA and sPLSA lies in the modeling of the response variable c_d connected to document d. Under the sPLSA model, each document and response arises from the following generative process:

- For each word w in document d
  - Choose a topic $z_{dw} \sim$ *Multinomial*$(\theta_{\mathbf{d}})$
  - Choose a word $w \sim$ *Multinomial*$(\beta_{\mathbf{z_{dw}}})$
- Draw a response $c_d \sim \mathcal{N}({\theta_{\mathbf{d}}}^T \mathbf{v}, \sigma^2)$

  Here the response comes from a Gaussian linear model. The mean is the inner product of topic distribution $\boldsymbol{\theta_d}$ and coefficient parameter vector $\boldsymbol{v}$.

Figure 1 illustrates the graphical model representation of PLSA and sPLSA, respectively.

It is worth noting that our approach for modeling c_d is different from that of sLDA. sLDA approximates a response variable, which in our case is c_d, as a linear combination of the mean Z_{dn} values. sLDA represents each $Z_{dn} = k$ as an indicator vector of length K where the k^{th} position is set to 1 and the others are set to 0. sLDA evaluates the mean Z_{dn} by taking the mean value of the vectors, which is expressed as $\overline{\mathbf{z_d}} = \sum_{n=1}^{N_d} Z_{dn}$. In Section 4, we empirically show that using a linear combination of θ_{dk} instead of $\overline{z_{dk}}$ yields v_k values that better factorize the response of the latent topics.

3.3. Likelihood function

The likelihood function in supervised PLSA consists of two parts. The first part is the likelihood for observing all the words in the corpus, $\mathbf{W}$, given the topic distributions for the documents, $\boldsymbol{\Theta}$, and the word distributions for the topics, β. Mathematically, it is as follows:

$$P(\mathbf{W}|\boldsymbol{\Theta}, \beta) = \prod_{d=1}^{M} \prod_{w \in d} \left(\sum_{k=1}^{K} \theta_{dk} \beta_{kw} \right)^{n(d,w)} \tag{1}$$

where $n(d, w)$ is the number of times word w appears in document d. Therefore, the log likelihood of the observed words is

$$J_1(\mathbf{\Theta}, \beta) = \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \log \left(\sum_{k=1}^{K} \theta_{dk} \beta_{kw} \right) \tag{2}$$

The second part of the likelihood function comes from the likelihood of the response variable. As shown in the generative process, we assume a linear model with Gaussian noise for modeling the response c_d. Specifically, we express c_d as follows:

$$c_d \sim \mathcal{N}\left(\sum_{k=1}^{K} \theta_{dk} v_k, \sigma^2 \right) \tag{3}$$

where v_k is the coefficient for θ_{dk}. The expression indicates that c_d is a random variable drawn from a Gaussian distribution with a mean $\sum_{k=1}^{K} \theta_{dk} v_k$ and a variance σ^2. The likelihood of observing all the responses is as follows:

$$P(\mathbf{c}|\mathbf{\Theta}, \mathbf{v}) = \prod_{d=1}^{M} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(- \frac{(c_d - \sum_{k=1}^{K} \theta_{dk} v_k)^2}{2\sigma^2} \right) \tag{4}$$

where $\mathbf{c}$ is a vector of all c_d in the corpus, and $\mathbf{v}$ is a vector of all v_k. v_k can be viewed as the contribution of topic k to the overall response. That is the higher a v_k value is the more its latent topic contributes to the response variable.

We assume a Gaussian prior on the coefficients v_k, i.e., $v_k \sim \mathcal{N}(0, \eta^2)$, which is equivalent to L_2 norm regularization. By ignoring some constants which do not impact the outcome of the likelihood maximization, the log likelihood of observing all the responses can be specified as:

$$J_2(\mathbf{\Theta}, \mathbf{v}) = -\sum_{d=1}^{M} \frac{(c_d - \sum_{k=1}^{K} \theta_{dk} v_k)^2}{2\sigma^2} - \frac{1}{2\eta^2} \sum_{k=1}^{K} v_k^2 \tag{5}$$

Equations (2) and (5) share $\mathbf{\Theta}$ as a parameter to estimate. This means we will need to unify both likelihoods into a single unified likelihood equation in order to estimate $\mathbf{\Theta}$. We accomplish this by normalizing the two likelihoods, and then linearly combining Eqs (2) and (5) as follows:

$$J(\mathbf{\Theta}, \beta, \mathbf{v}) = (1 - \lambda) \frac{J_1(\mathbf{\Theta}, \beta)}{\sum_{d=1}^{M} \sum_{w \in d} n(d, w)} + \lambda \frac{J_2(\mathbf{\Theta}, \mathbf{v})}{M} \tag{6}$$

where λ is a weighing constant and is a real number $\lambda \in [0, 1]$. Its value affects the perplexity of the latent topics, β, inferred by the unified likelihood. We discuss this in Section 4.

3.4. Parameter estimation

Now that we have established the unified likelihood, we can use it to derive formulas for iteratively updating the parameters $\boldsymbol{v}$, $\boldsymbol{\beta}$, and $\mathbf{\Theta}$ in order to converge the likelihood to its maximum value. At a high-level, we iteratively update the parameters one at a time until the likelihood converges. We illustrate the process in Fig. 2.

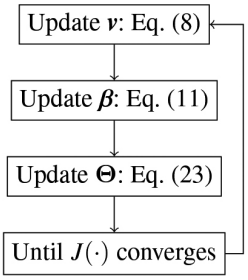


Fig. 2. The iterative updates of the parameter estimation process.

3.4.1. Updating $\mathbf{v}$

The values of $\mathbf{v}$ are only found in the second term of the unified likelihood. This means we can simply use $J_2(\boldsymbol{\Theta}, \mathbf{v})$ as the maximization objective to update $\mathbf{v}$ while fixing $\boldsymbol{\Theta}$ and β. If we use vector and matrix representations, maximizing $J_2(\boldsymbol{\Theta}, \mathbf{v})$ is equivalent to minimizing the following objective:

$$\min_{\mathbf{v}} \quad (\mathbf{c} - \boldsymbol{\Theta}\mathbf{v})^T(\mathbf{c} - \boldsymbol{\Theta}\mathbf{v}) + \frac{\sigma^2}{\eta^2}\mathbf{v}^T\mathbf{v} \tag{7}$$

It can be seen that the above objective function is strictly convex in v by its positive second derivative. By taking the first derivative of the function with respective to v and setting it to zero, we can obtain the analytic solution to $\mathbf{v}$ as follows

$$\mathbf{v} = \left(\boldsymbol{\Theta}^{\mathbf{T}}\boldsymbol{\Theta} + \frac{\sigma^2}{\eta^2}\mathbf{I}\right)^{-1}\boldsymbol{\Theta}^{\mathbf{T}}\mathbf{c} \tag{8}$$

This solution is equivalent to Ridge Regression or Tikhonov regularization [69].

3.4.2. Updating β

The values of β are only found in the first term of the unified likelihood. This means we can simply use $J_1(\boldsymbol{\Theta}, \beta)$ as the maximization objective. Similar to PLSA, we can use the EM algorithm to update β while fixing $\boldsymbol{\Theta}$ and $\mathbf{v}$. In the E-step, we apply Bayes' theorem and estimate the posterior probability of the topic k based on current parameters as follows:

$$P(k|d, w) = \frac{\theta_{dk}\beta_{kw}}{\sum_{k'=1}^{K}\theta_{dk'}\beta_{k'w}} \tag{9}$$

In the M-step, we maximize the expected complete data log-likelihood as follows:

$$\max_{\beta} E(J_1) = \sum_{d=1}^{M}\sum_{w\in d} n(d, w)\sum_{k=1}^{K} P(k|d, w)\log(\theta_{dk}\beta_{kw}) \tag{10}$$

with the constraint of $\sum_{w\in d}\beta_{kw} = 1$. Here $P(k|d, w)$ is obtained from the E-step. By using the Lagrange multiplier method to solve the constrained optimization problem in Eq. (10), we obtain the following update rule for β:

$$\beta_{kw} = \frac{\sum_{d=1}^{M} n(d, w)P(k|d, w)}{\sum_{d=1}^{M}\sum_{w\in d} n(d, w)P(k|d, w)} \tag{11}$$

3.4.3. Updating $\boldsymbol{\Theta}$

The values of $\boldsymbol{\Theta}$ are found in both terms of the unified likelihood. However, it is difficult to maximize $J(\boldsymbol{\Theta}, \beta, \mathbf{v})$ with respect to $\boldsymbol{\Theta}$, because $J_1(\boldsymbol{\Theta}, \beta)$ has a log of sums term that contains θ_{dk}. Instead of using $J_1(\boldsymbol{\Theta}, \beta)$ in $J(\boldsymbol{\Theta}, \beta, \mathbf{v})$, we use the same lower bound objective that the EM algorithm uses to approximate $J_1(\boldsymbol{\Theta}, \beta)$, which is derived as follows:

$$\begin{aligned}
J_1(\boldsymbol{\Theta}, \beta) &= \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \log \left(\sum_{k=1}^{K} \theta_{dk} \beta_{kw} \right) \\
&= \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \log \left(\sum_{k=1}^{K} P(k|d, w) \frac{\theta_{dk} \beta_{kw}}{P(k|d, w)} \right) \\
&\geqslant \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \sum_{k=1}^{K} P(k|d, w) \log \left(\frac{\theta_{dk} \beta_{kw}}{P(k|d, w)} \right) \\
&= \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \sum_{k=1}^{K} P(k|d, w) \log(\theta_{dk}) + \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \sum_{k=1}^{K} P(k|d, w) \log(\beta_{kw}) \\
&- \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \sum_{k=1}^{K} P(k|d, w) \log \big(P(k|d, w) \big)
\end{aligned} \tag{12}$$

Since the second and third terms in the above lower bound are constants with respect to $\boldsymbol{\Theta}$, we can drop them to obtain a simpler lower bound objective for optimizing $\boldsymbol{\Theta}$. The objective is as follows:

$$J_3(\boldsymbol{\Theta}) = \sum_{d=1}^{M} \sum_{w \in d} n(d, w) \sum_{k=1}^{K} P(k|d, w) \log(\theta_{dk}) \tag{13}$$

This means we use the following objective instead of the unified likelihood to update $\boldsymbol{\Theta}$:

$$J_L(\boldsymbol{\Theta}, \mathbf{v}) = (1 - \lambda) \frac{J_3(\boldsymbol{\Theta})}{\sum_{d=1}^{M} \sum_{w \in d} n(d, w)} + \lambda \frac{J_2(\boldsymbol{\Theta}, \mathbf{v})}{M} \tag{14}$$

The above objective is a concave function with respect to $\boldsymbol{\Theta}$ when we fix $\mathbf{v}$. We can solve for the values of $\boldsymbol{\Theta}$ that maximize the objective provided that the following constraint is met for every document d.

$$\sum_{k=1}^{K} \theta_{dk} = 1, \quad \forall\, d \in \mathcal{D} \tag{15}$$

The constraint must be met because each $\boldsymbol{\theta_d}$ is a probability distribution. However, this constraint results in a constrained optimization problem that is hard to solve with a simple closed form expression similar to the constrained optimization problem for estimating β_{kw} (Eq. (11)). This is because the gradient of $J_L(\cdot)$ with respect to θ_{dk} (Eq. (25)) yields an expression that consists of all the θ_{dk} parameters for all documents and the given k. This makes finding a closed form solution for θ_{dk} difficult. To overcome this difficulty, we transform the constrained optimization problem to an unconstrained optimization problem by expressing θ_{dk} in terms of a parameter τ_{dk} as follows:

$$\theta_{dk} = \text{SOFTMAX}(\tau_{dk}) \tag{16}$$

where $\tau_{dk} \in \mathbb{R}$ and SOFTMAX$(\cdot)$ is defined as follows:

$$\text{SOFTMAX}(\tau_{dk}) = \frac{\exp(\tau_{dk})}{\sum_{k'=1}^{K} \exp(\tau_{dk'})} \tag{17}$$

Irrespective of the value of τ_{dk}, SOFTMAX(τ_{dk}) returns a value in the range of $[0, 1]$ and the sum of all SOFTMAX(τ_{dk}) for each $\tau_{dk} \in \boldsymbol{\tau_d}$ is always 1. As a result, expressing θ_{dk} in terms of SOFTMAX(τ_{dk}) innately allows θ_{dk} to satisfy the constraint, and effectively transforms the constrained optimization to an unconstrained optimization problem.

Furthermore, we can reduce the number of τ_{dk} parameters from K to $K-1$, because one τ_{dk} is redundant since:

$$\theta_{dk} = \text{SOFTMAX}(\tau_{dk}) = 1 - \sum_{k'=1}^{K-1} \text{SOFTMAX}(\tau_{dk'}) \tag{18}$$

when $k = K$. To remove the redundant parameter, we note that subtracting a value from all τ_{dk} does not change the value of SOFTMAX$(.)$:

$$\begin{aligned} \text{SOFTMAX}(\tau_{dk} - h) &= \frac{\exp(\tau_{dk} - h)}{\sum_{k'=1}^{K} \exp(\tau_{dk'} - h)} \\ &= \frac{\exp(\tau_{dk}) \exp(-h)}{\sum_{k'=1}^{K} \exp(\tau_{dk'}) \exp(-h)} \\ &= \frac{\exp(\tau_{dk})}{\sum_{k'=1}^{K} \exp(\tau_{dk'})} \\ &= \text{SOFTMAX}(\tau_{dk}) \end{aligned} \tag{19}$$

As a result, we can express τ_{dk} with an auxiliary parameter μ_{dk} that is as follows:

$$\mu_{dk} = \tau_{dk} - \tau_{dK} \tag{20}$$

This results in $\mu_{dK} = 0$, which eliminates μ_{dK} for being an additional parameter of $\boldsymbol{\mu_d}$. Therefore, SOFTMAX$(.)$ simplifies to the following when $1 \leqslant k \leqslant K-1$:

$$\text{SOFTMAX}(\mu_{dk}) = \frac{\exp(\mu_{dk})}{1+\sum_{k'=1}^{K} \exp(\mu_{dk'})} \tag{21}$$

and to the following when $k = K$:

$$\text{SOFTMAX}(0) = \frac{1}{1 + \sum_{k'=1}^{K} \exp(\mu_{dk'})} \tag{22}$$

Finally, we can express θ_{dk} as follows in terms of $\boldsymbol{\mu_d}$:

$$\theta_{dk} = \begin{cases} \frac{\exp(\mu_{dk})}{1+\sum_{k'=1}^{K-1} \exp(\mu_{dk'})} & \text{if } 1 \leqslant k \leqslant K-1 \\ \frac{1}{1+\sum_{k'=1}^{K-1} \exp(\mu_{dk'})} & \text{if } k = K \end{cases} \tag{23}$$

The above representation of θ_{dk} ensures Eq. (15) holds. Therefore, instead of doing a constrained maximization with respect to $\boldsymbol{\Theta}$, we perform an unconstrained maximization with respect to μ.

We use the gradient ascent algorithm to maximize the objective function $J_L(\boldsymbol{\Theta}, \mathbf{v})$ in Eq. (14) with respect to μ by fixing $\mathbf{v}$. The partial derivative we use to update each μ_{dk} is as follows:

$$\frac{\partial J_L}{\partial \mu_{dk}} = \sum_{k'=1}^{K} \frac{\partial J_L}{\partial \theta_{dk'}} \frac{\partial \theta_{dk'}}{\partial \mu_{dk}} \tag{24}$$

where:

$$\frac{\partial J_L}{\partial \theta_{dk'}} = \frac{1-\lambda}{\sum_{d=1}^{M}\sum_{w\in d} n(d,w)} \frac{\sum_{w\in D} n(d,w)p(k'|d,w)}{\theta_{dk'}} + \frac{\lambda}{\sigma^2 M}\sum_{d=1}^{M}\left(c_d - \sum_{k=1}^{K}\theta_{dk}v_k\right)v_{k'} \quad (25)$$

and:

$$\frac{\partial \theta_{dk'}}{\partial \mu_{dk}} = \begin{cases} \theta_{dk}(1-\theta_{dk}) & \text{if } k' = K \\ -\theta_{dk'}\theta_{dk} & \text{if } k' \neq k \end{cases} \quad (26)$$

After we update each μ_{dk}, we update each θ_{dk} using Eq. (23).

3.5. Inference

After the parameter estimation is completed, we do the following to infer the latent topics and their factorized response values:

- We infer the latent topics from the topic-word distribution $\boldsymbol{\beta}$ by ranking the words for each latent topic k in descending order of the probability of the words belonging to the topic (β_{kw}). We then extract the most probable words of the topic to get an intuition about what each latent topic is about. We do this by analyzing the semantics of the extracted words.
- We infer the factorized response for each latent topic k from its v_k value. The larger the v_k value, the more dominant the topic is in determining the response variable.

4. Experiments

In this section, we discuss the dataset we used to test sPLSA, present experimental results, and compare our model to the baselines.[1]

4.1. Dataset

We tested sPLSA using bills which were placed for a vote in the United States Congress. The objective of our test is to generate the latent topics of the bills, and then rank them by controversy. We do this by first assigning a controversy score to each bill followed by inferring the factorized controversy score of each topic using sPLSA. We assign a controversy score to each bill by using the spread of the number of yes and no votes. The formula we use is as follows:

$$c_d = 1 - \frac{|a_d - b_d|}{a_d + b_d} \quad (27)$$

where c_d is the controversy score of bill d, a_d is the number of yes votes for the bill, and b_d is the number of no votes for the bill. A value of 0 indicates no controversy and occurs when the votes are either all yes or all no. A value of 1 indicates maximum controversy and occurs when the number of yes and no votes are evenly split. sPLSA uses the c_d value of the bills as the response variable and generates the latent topics of the bills. We use the v_k values generated by the model to rank the latent topics by controversy.

The reason why we selected congressional bills and their controversy scores as our dataset is to demonstrate applying sPLSA to a real world problem. Specifically, we want to identify contentious issues

[1]The dataset and source code for our experiments can be found at https://github.com/ealemayehu/splsa.

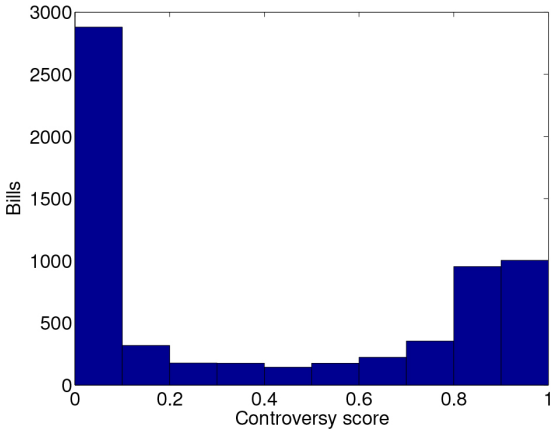


Fig. 3. Histogram of the distribution of the response variable calculated using Eq. (27).

in the United States Congress by generating their latent topics. By inferring their relative controversy using sPLSA, we can rank the topics by controversy, and identify the contentious issues by selecting the most controversial topics.

We collected bills starting from the 100^{th} Congress and ending with the 114^{th} Congress. This is for the years 1987 to early 2016. We used the Vote API of GovTrack [2] to obtain information about the votes. Next, we discarded the votes that are not associated with a bill. For votes that are associated with a bill, we kept the final votes for the bill. Finally, we obtained the digital content of the bills from the website of the U.S. Government Publishing Office.[3] We only obtained the content of bills that had a plain text version.

We were able to collect the votes and content of 6,403 bills. 5,531 bills were from the House of Representatives and 872 bills were from the Senate. 6,160 bills had more yes votes than no votes, and 243 bills had more no votes than yes votes. Figure 3 shows the distribution of the bills' controversy score.

We did the following preprocessing of the bills to create our dataset:

- Removed words which have characters that are not in the English alphabet.
- Removed words less than 4 characters in length.
- Removed common English words using Mallet's[4] stop-word list.
- Removed domain specific words using a custom stop-word list. The stop-word list has 157 words, and we created it by analyzing the word frequency of the bills. It mostly consists of legal terms.
- Selected the 15,000 most frequent words as the vocabulary of our corpus.

We then created the dataset as a bag-of-words representation of each bill.

4.2. Setup

We randomly partition our dataset as follows: 80% for training, 10% for validation, and 10% for testing. We initialize μ by sampling from a Gaussian distribution of mean 0 and variance 1. We initialize $\mathbf{\Theta}$ from the initial values of μ using Eq. (23). We initialize β by sampling from a uniform distribution and then

[2] https://www.govtrack.us.
[3] http://www.gpo.gov/fdsys.
[4] http://mallet.cs.umass.edu/.

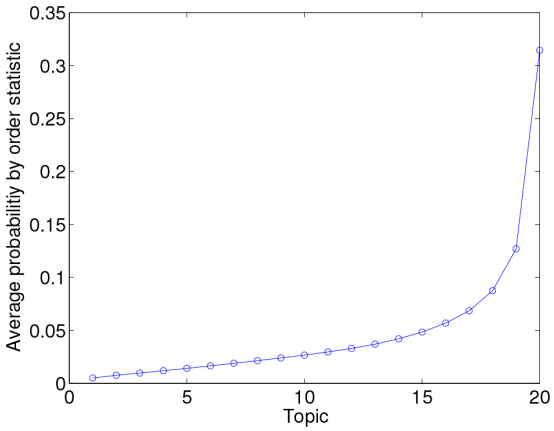


Fig. 4. The sparsity of $\theta_{\mathbf{d}}$ by plotting the average probability of the k^{th} order statistic of the topics in all $\theta_{\mathbf{d}}$ for $K = 20$ and $\lambda = 0.5$.

normalizing each $\beta_{\mathbf{k}}$, the word distribution of topic k, so that it becomes a valid probability distribution. We initialize $\mathbf{v}$ by setting all $v_k = 0$. We initialize values for λ, K, and η depending on the experiment we are running. Our implementation of sPLSA iteratively updates in lockstep $\mathbf{v}$, β, and μ until the unified likelihood converges. At the beginning of each iteration, we update $\mathbf{\Theta}$ using Eq. (23).

4.3. Evaluation metrics

We test a trained model by folding-in the test dataset similar to the way specified in [1]. This is essentially the same as training the model with the test dataset except β and $\mathbf{v}$ are not updated, and their values are obtained from the trained model. The only parameter we estimate in the folding-in process is $\mathbf{\Theta}$. We evaluate the performance of the model with test data as follows:

- For $\mathbf{\Theta}$ and β, we use the perplexity of the topics inferred from the test dataset. The lower the perplexity is, the better the values of $\mathbf{\Theta}$ and β.
- For $\mathbf{v}$, we use Pearson correlation to correlate each v_k with the average controversy score of the bills which have k as their most probable topic (s_k). s_k is calculated as follows:

$$s_k = \frac{\sum_d 1\{\max \theta_{\mathbf{d}} = \theta_{dk}\} c_d}{\sum_d 1\{\max \theta_{\mathbf{d}} = \theta_{dk}\}} \tag{28}$$

  The higher the correlation between $\mathbf{v}$ and $\mathbf{s}$ is, the better the values of $\mathbf{v}$, and the better $\mathbf{v}$ represents the relative controversy between the topics.

The reason why we can correlate each v_k with s_k is because the $\theta_{\mathbf{d}}$ are sparse. An example of the sparsity is illustrated in Fig. 4 where we aggregate the average probability of the k^{th} order statistic of the topics in all $\theta_{\mathbf{d}}$ for $K = 20$ and $\lambda = 0.5$. We can clearly see from the plot that the 20$^{\text{th}}$ order statistic is by far the most dominant topic.

4.4. Results

Table 2 shows the top 5 words for the 1$^{\text{st}}$, 2$^{\text{nd}}$, median, 2$^{\text{nd}}$ last, and last controversial topics for four experiments. Each experiment selected a unique $K \in \{10, 20, 30, 40\}$, and all the experiments set $\lambda = 0.5$ and $\eta = 1$. As we will see later, our choice of λ and η are optimal for our dataset. In addition to the top 5 words, the table shows the v_k coefficient of each topic. As we mentioned earlier, the v_k

Table 2
The top 5 words for the 1^{st}, 2^{nd}, median, 2^{nd} last, and last most controversial topics of selected K values as well as the v_k values of the topics

Rank	$K = 10$	$K = 20$	$K = 30$	$K = 40$
1	$v = 1.36$	$v = 1.886$	$v = 1.942$	$v = 2.527$
	Fundraising	Fisa	Alien	Educating
	Expense	Buddhist	Immigrants	Institutes
	Fisa	Outdoor	Secures	High
	Alien	Functions	Employing	Struggle
	Secures	Loaded	Stationing	Structuring
2	$v = 1.145$	$v = 1.268$	$v = 1.885$	$v = 2.262$
	Fundraising	Fundraising	Fisa	Payers
	Fisa	Expense	Buddhist	Fisa
	Buddhist	Relocates	Expense	Indispensable
	Expense	Appropriations	Fundraising	Paygo
	Appropriations	Expendable	Functions	Plain
$\lfloor \frac{K}{2} \rfloor$	$v = -0.508$	$v = -0.251$	$v = 0.229$	$v = 0.106$
	Defending	Finances	Foregone	Education
	Milestone	Commissary	Internal	Schofield
	Fisa	Board	Secures	Lobbying
	Forced	Persian	Nation	Educating
	Forbs	Companionship	Verifying	Childless
$K - 1$	$v = -0.844$	$v = -1.316$	$v = -2.381$	$v = -2.188$
	Plain	Propene	Defending	Chances
	Propene	Therapeutics	Milestone	Header
	Lancaster	Chaplains	Proclaimed	Chaplains
	Mammography	Lien	Forbs	Sequester
	Tarp	Fisa	Researcher	Frederick
K	$v = -0.977$	$v = -1.829$	$v = -3.604$	$v = -2.569$
	Healing	Drowning	Commissary	Houses
	Houses	Prison	Safeguarding	Expense
	Fundraising	Bushel	Chaplains	Prohibited
	Secures	Mammography	Vessel	Amounts
	Payers	Sttr	Transnational	Distributors

values estimate the controversy of the topics and we use them to select the topics shown in the table. For $K = 10$, we find that some of the topics overlap. For example, the words "fundraising", "expense" and "fisa" (Foreign Intelligence Surveillance) appear in multiple topics. This is because the number of topics is insufficient. On the other hand, $K = 40$ has more granular topics that overlap less. For $K = 40$, we can infer from the words that the 1^{st} topic is about higher education, the 2^{nd} is about funding, the 3^{rd} is about child education, the 4^{th} is either about religion or sequester related budget cuts, and the 5^{th} topic is about housing. We can therefore conclude that K has to be large enough in order to avoid topic overlap. We also observe that as the value of K increases so does the variance of the v_k values. This means that the most controversial topics of larger K values are more controversial than the most controversial topics of smaller K values. This makes intuitive sense since the overlap between the most controversial topics and other topics gets smaller as K increases.

4.4.1. Comparison to baseline

Our baseline is an sLDA model. The response variable for the model is the controversy score. We used the 'slda.em' function in the R "lda" package[5] to train the sLDA model using the training dataset.

[5]https://cran.r-project.org/web/packages/lda/lda.pdf.

Table 3
Comparison of the Pearson correlation between v_k and s_k for our model and sLDA

K	10	20	30	40
sPLSA	0.9837	0.9850	0.9632	0.9182
sLDA	0.9530	0.8427	0.7550	0.4614

We run the model for each $K \in \{10, 20, 30, 40\}$ by setting $\alpha = 0.1$, $\beta = 0.1$, and variance $= 0.25$. We then correlated the v_k and s_k values of sLDA and compared the correlation to that of our model when $\lambda = 0.5$ and $\eta = 1$. For a fair comparison, we used the training data to evaluate the s_k values. This is because unlike sPLSA, sLDA does not have access to the response variable when using test data, since its purpose is to predict the response variable. Table 3 illustrates the comparison. From the table, we clearly see our model correlates significantly better as K increases in value. This is the case because the v_k values of sPLSA are trained on the topic distributions, $\theta_{\mathbf{d}}$, whereas the equivalent coefficients in sLDA are trained on the realized topic distributions. For sLDA, the variance between the $\theta_{\mathbf{d}}$ and the realized topic distributions significantly increases as K increases, and this deteriorates the ability of the sLDA coefficients in approximating the controversy of the $\theta_{\mathbf{d}}$.

sPLSA is designed for topic discovery and latent response inference. This comes at the expense of its prediction performance. Theoretically, we can use sPLSA in a semi-supervised setting where we mix both labeled and unlabeled data, and then try to predict the labels for the unlabeled data. In such a scenario, we update $\mathbf{v}$ using the labeled data, β using both the labeled and unlabeled data, and $\mathbf{\Theta}$ using both the labeled and unlabeled data. However, for the unlabeled data, we update $\mathbf{\Theta}$ by setting $\lambda = 0$, since we do not have a response value. Once we train our model, we linearly combine the $\theta_{\mathbf{d}}$ values of the unlabeled data with the $\mathbf{v}$ values to predict the labels. Figure 5 shows the RMSE values of our model's predictions versus the RMSE values of the sLDA predictions. Clearly, we can see that the RMSE values of our model are significantly worse than the RMSE values of sLDA. The reason why our model performs weakly is because it uses $\theta_{\mathbf{d}}$ values to do the prediction. The $\theta_{\mathbf{d}}$ values are the average estimate for the topic distributions of the words in each document. In the case of sLDA, the realized topic distributions of the words, Z_d, is used.

4.4.2. Efficiency

We run sPLSA and sLDA on a MacPro laptop with a 2 GHz processor and 16 GB RAM on the training dataset for various values of K. Figure 6 compares the training time of sPLSA with sLDA for various values of K. As we can see, the training time of sPLSA was at least 6 times faster than sLDA. This is the case because the EM algorithm used by sPLSA converges much faster than the Gibbs sampling used by the sLDA implementation. This is despite the fact that our implementation is a single-threaded Java program not optimized for efficiency while the core of sLDA is efficiently implemented in C.

The reason why Gibbs sampling converges a lot slower than the EM algorithm is because the topics tend to depend on one another. This prolongs the burn-in period for the Gibbs sampling process where a stationary distribution has not been achieved. A stationary distribution needs to be achieved for the actual sampling to take place. During the burn-in period, the Gibbs sampling process can diverge at times. On the other hand, EM does not have the equivalent of a burn-in period and every iteration of the algorithm is guaranteed to monotonically improve the convergence of the likelihood.

4.4.3. Impact of η

We trained the model with $K = 20$ and $\lambda = 0.5$ for each $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. We then tested each model with the validation dataset, and obtained the results shown in Table 4. Overall, we can see from the table $\eta = 1$ yields the best results.

Table 4
The perplexity and Pearson correlation values on the validation dataset for different values of η

η	0.001	0.01	0.1	1	10	100	1000
Perplexity	2072.6	2307.3	2062.4	2102.3	2056.7	2151.2	2053.5
Correlation	0.335	−0.302	0.982	0.987	0.951	0.966	0.976

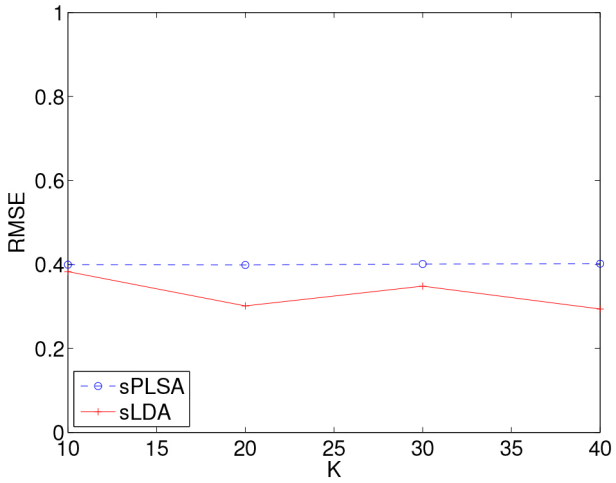


Fig. 5. The prediction RMSE values of sPLSA and sLDA at various values of K.

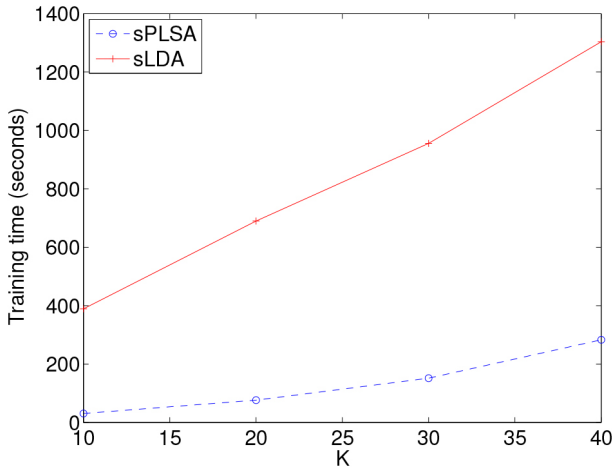


Fig. 6. Comparison of the training time of sPLSA and sLDA for various values of K.

4.4.4. Impact of λ

We trained the model for each combination of $K \in \{10, 20, 30, 40\}$ and λ from 0 to 1 in increments of 0.1. We then tested the model using the test dataset. Figure 7 shows the perplexity values for the various combinations of K and λ.

From the figure, we can generally see that as λ increases the perplexity increases as well. For smaller K values this increase is noisy, but for larger K values it gets smoother. The increase in perplexity accelerates as λ approaches 1. We also notice that as K gets larger the overall perplexity gets lower.

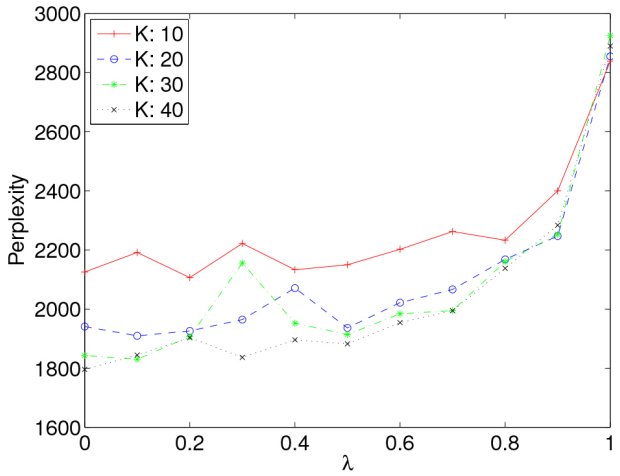


Fig. 7. The perplexity for various combinations of K and λ.

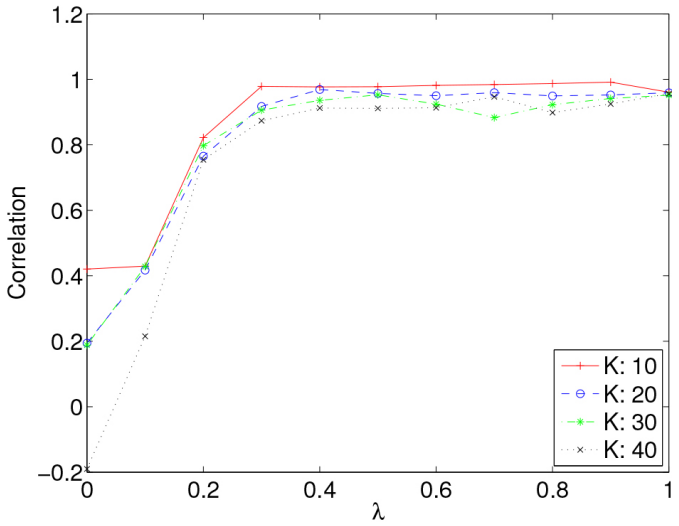


Fig. 8. The Pearson correlation between v_k and s_k for various combinations of K and λ.

Figure 8 shows the values of the Pearson correlation between v_k and s_k for the various combinations of K and λ.

From the figure, we can generally see that the correlation increases steeply from $\lambda = 0$ to approximately $\lambda = 0.3$. It then decelerates rapidly and levels off to within a noisy range. We can conclude from Figs 7 and 8 that for a fixed K improving the perplexity by increasing λ generally deteriorates the correlation and vice-versa. However, there is a range of λ values between 0.4 and 0.5 where the perplexity is not that far from the lowest perplexity and the correlation is not much different from the maximum correlation. Our ideal λ is therefore in the range of [0.4, 0.5] for our dataset.

4.4.5. Sample topics

Tables 5, 6, and 7 show the top words for the topics generated by PLSA, sLDA, and sPLSA. In general, we can see that very similar topics are generated by all three models. For example, topic 7 for PLSA,

Table 5
Top 5 words for the topics of PLSA when $K = 10$

1	Transnational	Alien	Finances	Board	Persian
2	Fisa	Buddhist	Outdoor	Loaded	Healing
3	Washoe	Prohibited	Enemy	Lancaster	Mammography
4	Healing	Plain	Indispensable	Cards	Chiefs
5	Defending	Milestone	Fisa	Forbs	Forced
6	Plain	Houses	Inclusive	Indispensable	Propene
7	Education	Educating	Lobbying	Schofield	Fundraising
8	Secures	Intell	Directly	Foregone	Rescind
9	Fundraising	Expense	Appropriations	Fisa	Relocates
10	Defending	Fisa	Fundraising	Milestone	Plain

Table 6
Top 5 words for the topics of sLDA when $K = 10$

1	Foregone	Internally	Securitization	Nationally	Countries
2	Educating	Education	Lobbyists	Scholars	Childless
3	Fundraising	Blvd	Subsystem	Administrator	Entitles
4	Lance	Wastewater	Enemy	Conservancy	Prohibition
5	Defending	Milford	Forbs	Forced	Armstrong
6	Authorities	Constructing	Traineeships	Subsystem	Systematically
7	Plains	Healing	Paying	Cards	Inclusive
8	Persistent	Commissary	Coursework	Finances	Attitudes
9	Fundraising	Expense	Relocations	Transplantation	Appropriation
10	Fisa	Buddhist	Fundraising	Reseller	Securitization

Table 7
Top 5 words for the topics of sPLSA when $K = 10$

1	Fundraising	Expense	Fisa	Alien	Secures
2	Healing	Houses	Fundraising	Secures	Payers
3	Finances	Companionship	Commissary	Bank	Lobbying
4	Transnational	Prohibited	Enemy	Washoe	Synthetic
5	Persian	Font	Loaded	Agricultural	Eligibility
6	Educating	Healing	Lobbying	Education	Fisa
7	Plain	Propene	Lancaster	Mammography	Tarp
8	Fundraising	Fisa	Buddhist	Expense	Appropriations
9	Defending	Milestone	Fisa	Forced	Forbs
10	Fisa	Healing	Plain	Secures	Transnational

topic 2 for sLDA, and topic 6 for sPLSA are about education. This illustrates that the perplexity trade-off we did in selecting $\lambda = 0.5$ did not adversely affect the quality of the topics generated by sPLSA.

4.5. Case study

For each topic listed in Table 2 where $K = 40$, we sampled the bill which has the highest probability for the topic. We summarize the bills and analyze their connectedness to their corresponding topics in Tables 8, 9, 10, 11, and 12. As we can see from the tables, the controversy score of the bills closely aligns with the controversy level of the topics. In addition, the themes of the topics we specified in the beginning of Section 4.4 partially or fully match the theme of the bills with the exception of the bill for the second least controversial topic. This is primarily because the theme of the second least controversial topic is hard to determine based on the top words of the topic.

Table 8
Sample bill for the most controversial topic

Bill ID	H.R. 609
Title	College Access and Opportunity Act.
Year	2006
Yes Votes	221
No Votes	199
Controversy Score	0.95
Topic Probability	0.52
Description	This bill is about higher education, and amends the Higher Education Act of 1965.
Analysis	The controversy score is on the high-end and the theme of the bill, higher education, matches that of the topic.

Table 9
Sample bill for the second most controversial topic

Bill ID	H.R. 2491
Title	Budget Reconciliation Act of 1995
Year	1995
Yes Votes	235
No Votes	192
Controversy Score	0.90
Topic Probability	0.50
Description	This bill is about the federal budget for 1996.
Analysis	The controversy score is close to the high-end, and the theme of the bill, funding, matches that of the topic.

Table 10
Sample bill for the most moderately controversial topic

Bill ID	H.R. 2
Title	Student Results Act of 1999
Year	1999
Yes Votes	358
No Votes	67
Controversy Score	0.31
Topic Probability	0.91
Description	This bill is about child education.
Analysis	The controversy score is in the middle range, and the theme of the bill, child education, matches that of the topic.

Table 11
Sample bill for the second least controversial topic

Bill ID	S. RES. 501
Title	A resolution honoring the sacrifice of the members of the United States Armed Forces who have been killed in Iraq and Afghanistan.
Year	2008
Yes Votes	95
No Votes	0
Controversy Score	0.00
Topic Probability	0.78
Description	As the title indicates this bill is a resolution honoring servicemen killed in combat.
Analysis	The controversy score is the lowest possible. However, it is hard to align the theme of the bill with that of the topic.

Table 12
Sample bill for the least controversial topic

Bill ID	H.R. 2158
Title	Departments of Veterans Affairs and Housing and Urban Development, and Independent Agencies Appropriations Act, 1998
Year	1998
Yes Votes	397
No Votes	31
Controversy Score	0.10
Topic Probability	0.34
Description	This bill is about benefits to veterans. Among the benefits is a program account to fund veterans housing benefits.
Analysis	The controversy score is close to the low end. Partially, the theme of the bill matches that of the topic.

5. Conclusion and future work

In this paper, we introduce sPLSA. We describe sPLSA as an extension of PLSA that is an analog of what sLDA is to LDA. Similar to sLDA, sPLSA processes a response variable associated with the documents to factorize the responses on a per-topic basis. We discuss the advantage sPLSA has over sLDA for doing latent response analysis such as the ranking of the topics by their factorized responses and the execution efficiency of the model. In addition, we discuss the advantage sLDA has over sPLSA for predicting the responses of documents. We experimentally demonstrated sPLSA on a real world problem by doing a latent controversy analysis of topics inferred from the bills of the United States Congress.

This work is an initial step towards a promising research direction. The presented model assumes the response comes from a Gaussian linear model. This assumption can be relaxed by extending the distribution of the response to a generalized linear model (GLM) [70], which allows for response variables that have error distribution models other than a Gaussian distribution. In future work, we plan to extend sPLSA to other types of response variables including the multinomial, the Poisson, the gamma, Weibull, inverse Gaussian, and so on. This will allow us to apply sPLSA to do latent topic analysis on a more diverse set of problems. Last but not the least, we will explore to combine the proposed model with neural networks by leveraging their nonlinearity modeling capability and extend the work to the realm of neural topic models [71].

References

[1] T. Hofmann, Probabilistic latent semantic analysis, in: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.
[2] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 50–57.
[3] T. Hofmann, Latent semantic models for collaborative filtering, *ACM Transactions on Information Systems (TOIS)* **22**(1) (2004), 89–115.
[4] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman and W.T. Freeman, Discovering object categories in image collections, in: Proceedings of IEEE International Conference on Computer Vision, 2005, pp. 134–141.
[5] M. Hoffman, D. Blei and P.R. Cook, Finding latent sources in recorded music with a shift-invariant HDP, in: Proceedings of the conference on digital audio effects, 2009, pp. 121–128.
[6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, Indexing by latent semantic analysis, *Journal of the American Society for Information Science* **41**(6) (1990), 391.
[7] D.M. Blei, A.Y. Ng and M.I. Jordan, Latent dirichlet allocation, *The Journal of Machine Learning Research* **3** (2003), 993–1022.
[8] M. Girolami and A. Kabán, On an equivalence between PLSI and LDA, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, 2003, pp. 433–434.

[9] T.L. Griffiths and M. Steyvers, Finding scientific topics, *Proceedings of the National Academy of Sciences* **101**(Suppl 1) (2004), 5228–5235.
[10] V.-A. Nguyen, J.L. Boyd-Graber and P. Resnik, Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling., in: EMNLP, 2014, pp. 1752–1757.
[11] Y. Lu, Q. Mei and C. Zhai, Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA, *Information Retrieval* **14**(2) (2011), 178–203.
[12] J.D. Mcauliffe and D.M. Blei, Supervised topic models, in: Advances in neural information processing systems, 2008, pp. 121–128.
[13] T. Hofmann, The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data, in: IJCAI, Vol. 99, 1999, pp. 682–687.
[14] T. Hofmann, J. Puzicha and M.I. Jordan, Learning from dyadic data, *Advances in neural information processing systems* (1998), 466–472.
[15] T. Hofmann and J. Puzicha, Unsupervised Learning from Dyadic Data, *Technical Report* (1998), 1–32.
[16] C. Zhai, A. Velivelli and B. Yu, A cross-collection mixture model for comparative text mining, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 743–748.
[17] Q. Mei and C. Zhai, A mixture model for contextual text mining, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 649–655.
[18] Q. Mei, D. Cai, D. Zhang and C. Zhai, Topic modeling with network regularization, in: Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 101–110.
[19] M. Rosen-Zvi, T. Griffiths, M. Steyvers and P. Smyth, The author-topic model for authors and documents, in: Proceedings of the 20th conference on Uncertainty in artificial intelligence, AUAI Press, 2004, pp. 487–494.
[20] T. Iwata, T. Hirao and N. Ueda, Probabilistic latent variable models for unsupervised many-to-many object matching, *Information Processing & Management* **52**(4) (2016), 682–697.
[21] I. Vulić, W. De Smet, J. Tang and M.-F. Moens, Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications, *Information Processing & Management* **51**(1) (2015), 111–147.
[22] D.M. Blei, Probabilistic topic models, *Communications of the ACM* **55**(4) (2012), 77–84.
[23] C. Wang, D. Blei and F.-F. Li, Simultaneous image classification and annotation, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1903–1910.
[24] S. Lacoste-Julien, F. Sha and M.I. Jordan, DiscLDA: Discriminative learning for dimensionality reduction and classification, in: Advances in neural information processing systems, 2009, pp. 897–904.
[25] J. Zhu, A. Ahmed and E.P. Xing, MedLDA: maximum margin supervised topic models for regression and classification, in: Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 1257–1264.
[26] S. Jameel, W. Lam and L. Bing, Supervised topic models with word order structure for document classification and retrieval learning, *Information Retrieval Journal* **18**(4) (2015), 283–330.
[27] D. Ramage, D. Hall, R. Nallapati and C.D. Manning, Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, Association for Computational Linguistics, 2009, pp. 248–256.
[28] M. Kar, S. Nunes and C. Ribeiro, Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model, *Information Processing & Management* **51**(6) (2015), 809–833.
[29] S. Park, W. Lee and I.-C. Moon, Associative topic models with numerical time series, *Information Processing & Management* **51**(5) (2015), 737–755.
[30] K. Seshadri, S.M. Shalinie and C. Kollengode, Design and evaluation of a parallel algorithm for inferring topic hierarchies, *Information Processing & Management* **51**(5) (2015), 662–676.
[31] F. Colace, M. De Santo, L. Greco and P. Napoletano, Weighted word pairs for query expansion, *Information Processing & Management* **51**(1) (2015), 179–193.
[32] E.B. Sudderth, A. Torralba, W.T. Freeman and A.S. Willsky, Learning hierarchical models of scenes, objects, and parts, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2, IEEE, 2005, pp. 1331–1338.
[33] L.-J. Li, R. Socher and L. Fei-Fei, Towards total scene understanding: Classification, annotation and segmentation in an automatic framework, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2036–2043.
[34] D. Mimno and A. McCallum, Topic models conditioned on arbitrary features with dirichlet-multinomial regression, *International Conference on Uncertainty in Artificial Intelligence (UAI)* (2008).
[35] K. Thambiratnam and F. Seide, Learning spoken document similarity and recommendation using supervised probabilistic latent semantic analysis, in: INTERSPEECH, 2007, pp. 334–337.
[36] R. Fergus, L. Fei-Fei, P. Perona and A. Zisserman, Learning Object Categories from GoogleâĂŹs Image Search, in: Proceedings of IEEE International Conference on Computer Vision, 2005, pp. 234–241.
[37] T. Wang and C. Liu, Human Action Recognition Using Supervised pLSA, *International Journal of Signal Processing, Image Processing and Pattern Recognition* **6**(4) (2013), 403–414.

[38] D. Aliyanto, R. Sarno and B.S. Rintyarna, Supervised probabilistic latent semantic analysis (sPLSA) for estimating technology readiness level, in: 2017 11th International Conference on Information & Communication Technology and System (ICTS), IEEE, 2017, pp. 79–84.
[39] R. Salakhutdinov and G. Hinton, Deep boltzmann machines, in: Artificial intelligence and statistics, PMLR, 2009, pp. 448–455.
[40] H. Larochelle and S. Lauly, A neural autoregressive topic model, *Advances in Neural Information Processing Systems* **25** (2012), 2708–2716.
[41] D.P. Kingma and M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114* (2013).
[42] Z. Cao, S. Li, Y. Liu, W. Li and H. Ji, A novel neural topic model and its supervised extension, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29, 2015.
[43] C.E. Moody, Mixing dirichlet topic models and word embeddings to make lda2vec, *arXiv preprint arXiv:1605.02019* (2016).
[44] A.B. Dieng, C. Wang, J. Gao and J. Paisley, Topicrnn: A recurrent neural network with long-range semantic dependency, *arXiv preprint arXiv:1611.01702* (2016).
[45] P. Gupta, Y. Chaudhary, F. Buettner and H. Schütze, texttovec: Deep contextualized neural autoregressive models of language with distributed compositional prior, *International Conference on Learning Representation* (2019).
[46] R. Murakami and B. Chakraborty, Investigating the Efficient Use of Word Embedding with Neural-Topic Models for Interpretable Topics from Short Texts, *Sensors* **22**(3) (2022), 852.
[47] M. Grootendorst, BERTopic: Neural topic modeling with a class-based TF-IDF procedure, *arXiv preprint arXiv:2203.05794* (2022).
[48] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du and W. Buntine, Topic Modelling Meets Deep Neural Networks: A Survey, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), 2021.
[49] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat and A. Hassan, Topic modeling algorithms and applications: A survey, *Information Systems* (2022), 102131.
[50] K.K. Ladha, A spatial model of legislative voting with perceptual error, *Public Choice* **68**(1–3) (1991), 151–174.
[51] J. Londregan, Estimating legislators' preferred points, *Political Analysis* **8**(1) (1999), 35–56.
[52] G.W. Cox and K.T. Poole, On measuring partisanship in roll-call voting: The US House of Representatives, 1877-1999, *American Journal of Political Science* (2002), 477–489.
[53] J. Clinton, S. Jackman and D. Rivers, The statistical analysis of roll call data, *American Political Science Review* **98**(2) (2004), 355–370.
[54] M. Thomas, B. Pang and L. Lee, Get out the vote: Determining support or opposition from Congressional floor-debate transcripts, in: Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics, 2006, pp. 327–335.
[55] S. Gerrish and D.M. Blei, Predicting legislative roll calls from text, in: Proceedings of the 28th international conference on machine learning (icml-11), 2011, pp. 489–496.
[56] S. Gerrish and D.M. Blei, How they vote: Issue-adjusted models of legislative behavior, in: Advances in Neural Information Processing Systems, 2012, pp. 2753–2761.
[57] Y. Fang, L. Si, N. Somasundaram and Z. Yu, Mining contrastive opinions on political texts using cross-perspective topic model, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, 2012, pp. 63–72.
[58] Y. Gu, Y. Sun, N. Jiang, B. Wang and T. Chen, Topic-factorized ideal point estimation model for legislative voting network, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 183–192.
[59] C. Chen, F. Ibekwe-SanJuan, E. SanJuan and C. Weaver, Visual analysis of conflicting opinions, in: Visual Analytics Science And Technology, 2006 IEEE Symposium On, IEEE, 2006, pp. 59–66.
[60] M. Tsytsarau, T. Palpanas and K. Denecke, Scalable discovery of contradictions on the web, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 1195–1196.
[61] W.-H. Lin, T. Wilson, J. Wiebe and A. Hauptmann, Which side are you on?: identifying perspectives at the document and sentence levels, in: Proceedings of the Tenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2006, pp. 109–116.
[62] S. Somasundaran and J. Wiebe, Recognizing stances in online debates, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, Association for Computational Linguistics, 2009, pp. 226–234.
[63] J. Ashford, L. Turner, R. Whitaker, A. Preece, D. Felmlee and D. Towsley, Understanding the signature of controversial Wikipedia articles through motifs in editor revision networks, in: Companion Proceedings of the 2019 World Wide Web Conference, 2019, pp. 1180–1187.
[64] K. Kanclerz, A. Figas, M. Gruza, T. Kajdanowicz, J. Kocoń, D. Puchalska and P. Kazienko, Controversy and conformity: from generalized to personalized aggressiveness detection, in: Proceedings of the 59th Annual Meeting of the Association

for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 5915–5926.
[65] D.A. Morris-O'Connor, A. Strotmann and D. Zhao, The colonization of Wikipedia: evidence from characteristic editing behaviors of warring camps, *Journal of Documentation* (2022).
[66] S. Benslimane, J. Azé, S. Bringay, M. Servajean and C. Mollevi, Controversy Detection: a Text and Graph Neural Network Based Approach, in: International Conference on Web Information Systems Engineering, Springer, 2021, pp. 339–354.
[67] E.E. Küçük, S. Takır and D. Küçük, Controversy detection on health-related tweets, in: Proceedings of the 14th International Symposium on Health Informatics and Bioinformatics, 2021, p. 60.
[68] K. Garimella, G.D.F. Morales, A. Gionis and M. Mathioudakis, Quantifying controversy on social media, *ACM Transactions on Social Computing* **1**(1) (2018), 1–27.
[69] A.E. Hoerl and R.W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1) (1970), 55–67.
[70] P. McCullagh and J.A. Nelder, Generalized linear models, Vol. 37, CRC press, 1989.
[71] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du and W. Buntine, Topic modelling meets deep neural networks: A survey, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), 2021, pp. 4713–4720.