Learning Document Embeddings Along With Their Uncertainties

Santosh Kesiraju, Oldřich Plchot, Lukáš Burget, and Suryakanth V. Gangashetty

***Abstract*—Majority of the text modeling techniques yield only point-estimates of document embeddings and lack in capturing the uncertainty of the estimates. These uncertainties give a notion of how well the embeddings represent a document. We present Bayesian subspace multinomial model (Bayesian SMM), a generative log-linear model that learns to represent documents in the form of Gaussian distributions, thereby encoding the uncertainty in its covariance. Additionally, in the proposed Bayesian SMM, we address a commonly encountered problem of intractability that appears during variational inference in mixed-logit models. We also present a generative Gaussian linear classifier for topic identification that exploits the uncertainty in document embeddings. Our intrinsic evaluation using perplexity measure shows that the proposed Bayesian SMM fits the unseen test data better as compared to the state-of-the-art neural variational document model on (*Fisher*) speech and (*20Newsgroups*) text corpora. Our topic identification experiments show that the proposed systems are robust to over-fitting on unseen test data. The topic ID results show that the proposed model outperforms state-of-the-art unsupervised topic models and achieve comparable results to the state-of-the-art fully supervised discriminative models.**

***Index Terms*—Bayesian methods, embeddings, topic identification.**

I. Introduction

LEARNING word and document embeddings have proven to be useful in a wide range of information retrieval, speech and natural language processing applications [1]–[5].

These embeddings elicit the latent semantic relations present among the co-occurring words in a sentence or *bag-of-words* from a document. Majority of the techniques for learning these embeddings are based on two complementary ideologies, (i) topic modeling, and (ii) word prediction. The former methods are primarily built on top of the *bag-of-words* model and tend to capture higher-level semantics such as topics. The latter techniques capture lower-level semantics by exploiting the contextual information of words in a sequence [6]–[8].

Manuscript received October 4, 2019; revised February 23, 2020 and May 25, 2020; accepted July 15, 2020. Date of publication July 27, 2020; date of current version August 14, 2020. This work was supported by the Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science LQ1602". The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfeng Gao *(Corresponding author: Santosh Kesiraju.)*

Santosh Kesiraju is with the Brno University of Technology and International Institute of Information Technology, Hyderabad 500032, India (e-mail: kesiraju@fit.vutbr.cz).

Oldřich Plchot and Lukáš Burget are with the Brno University of Technology, 61200, Czechia (e-mail: iplchot@fit.vutbr.cz; burget@fit.vutbr.cz).

Suryakanth V. Gangashetty is with the International Institute of Information Technology, Hyderabad 500032, India (e-mail: svg@iiit.ac.in).

Digital Object Identifier 10.1109/TASLP.2020.3012062

On the other hand, there is a growing interest towards developing pre-trained language models [9], [10], that are then fine-tuned for specific tasks such as document classification, question answering, named entity recognition, etc. Although these models achieve state-of-the-art results in several NLP tasks; they require enormous computational resources to train [11].

Latent variable models [12] are a popular choice in unsupervised learning; where the observed data is assumed to be generated through the latent variables according to a stochastic process. The goal is then to estimate the model parameters, and also the latent variables. In probabilistic topic models (PTMs), the latent variables are attributed to topics, and the generative process assumes that every document is a distribution over topics and every topic is modeled as a distribution over words in the vocabulary [13]. Recent works showed that auto-encoders could also be seen as generative models for images and text [14], [15]. Generative models allow us to incorporate prior information about the latent variables, and with the help of variational Bayes (VB) techniques [14], [16], [17], one can infer a posterior distribution over the latent variables instead of just point-estimates. The posterior distribution captures the uncertainty of the latent variable estimates while trying to explain (fit) the observed data and our prior belief. In the context of text modeling, these latent variables are seen as embeddings.

In this paper, we present the Bayesian subspace multinomial model (Bayesian SMM) as a generative model for the *bag-of-words* representation of documents. We show that our model can learn to represent each document in the form of a Gaussian distribution, thereby encoding the uncertainty in its covariance. Further, we propose a generative Gaussian classifier that exploits this uncertainty for topic identification (ID). The proposed VB framework can be extended in a straightforward way for subspace n-gram model [18], that can model n-gram distribution of words in sentences.

Earlier, (non-Bayesian) SMM was used for learning document embeddings in an unsupervised fashion. They were then used for training linear classifiers for topic ID from spoken and textual documents [19], [20]. However, one of the limitations was that the learned document embeddings (also termed as document i-vectors) were only point-estimates and were prone to over-fitting, especially for shorter documents. Our proposed model can overcome this problem by capturing the uncertainty of the embeddings in the form of posterior distributions.

Given the significant prior research in PTMs and related algorithms for learning representations, it is crucial to draw precise relations between the presented model and previous works. We

do this from the following viewpoints: (a) graphical models illustrating the dependency of random and observed variables, (b) assumptions of distributions over random variables and their limitations, and (c) approximations made during the inference and their consequences.

The contributions of this paper are as follows: (a) we present Bayesian subspace multinomial model and analyze its relation to popular models such as latent Dirichlet allocation (LDA) [21], correlated topic model (CTM) [22], paragraph vector (PV-DBOW) [8] and neural variational document model (NVDM) [15], (b) we adapt tricks from [14] for faster and efficient variational inference of the proposed model, (c) we combine optimization techniques from [23], [24] and use them to train the proposed model, (d) we propose a generative Gaussian classifier that exploits uncertainty in the posterior distribution of document embeddings, (e) we provide experimental results on both text and speech data showing that the proposed document representations achieve state-of-the-art perplexity scores, and (f) with our proposed classification systems, we illustrate robustness of the model to over-fitting and at the same time obtain superior classification results when compared systems based on state-of-the-art unsupervised models.

We begin with the description of Bayesian SMM in Section II, followed by VB for the model in Section III. The complete VB training procedure and algorithm is presented in Section III-A. The procedure for inferring the document embedding posterior distributions for (unseen) documents is described in Section III-B. Section IV presents a generative Gaussian classifier that exploits the uncertainty encoded in document embedding posterior distributions. Relationship between Bayesian SMM and existing popular topic models is described in Section V. Experimental details are given in Section VI, followed by results and analysis in Section VII. Finally, we conclude and discuss directions for future research in Section VIII.

II. Bayesian Subspace Multinomial Model

The following steps explain the generative process of our model. For each document, a K-dimensional latent vector $\boldsymbol{w}$ is generated from isotropic $\mathrm{Gaussian}$ prior with mean $\mathbf{0}$ and precision λ:

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w} \,|\, \mathbf{0}\,, (\lambda \boldsymbol{I})^{-1}) \tag{1}$$

The latent vector $\boldsymbol{w}$ is a low dimensional embedding ($K \ll V$) of document-specific distribution of words, where V is the size of the vocabulary. More precisely, for each document, the V-dimensional vector of word probabilities $\boldsymbol{\theta}$ is calculated as:

$$\boldsymbol{\eta} = \boldsymbol{m} + \boldsymbol{T}\,\boldsymbol{w}, \tag{2}$$

$$\boldsymbol{\theta} = \mathrm{softmax}(\boldsymbol{\eta}), \tag{3}$$

where $\{\boldsymbol{m}, \boldsymbol{T}\}$ are parameters of the model. The vector $\boldsymbol{m}$ known as universal background model (or bias) represents log uni-gram probabilities of words. $\boldsymbol{T}$ known as total variability (or weight) matrix [25], [26] is a low-rank matrix defining subspace of document-specific distributions.

Finally, for each document, a vector of word counts $\boldsymbol{x}$ (*bag-of-words*) is sampled from $\mathrm{Multinomial}$ distribution:

$$\boldsymbol{x} \sim \mathrm{Multi}(\boldsymbol{\theta}; N), \tag{4}$$

where N is the number of words in the document.

$\boldsymbol{\eta}$ from (3) represents the *natural parameters* of the Multinomial distribution. Further, we can see that our model is linear in the space of natural parameters (2). Note that the parameters of any probability distribution under the *exponential family* can be expressed in terms of its *natural parameters* [16].

The above described generative process fully defines our Bayesian model, which we will now use to address the following problems: given training data $\boldsymbol{X}$, we estimate model parameters $\{\boldsymbol{m}, \boldsymbol{T}\}$ and, for any given document $\widehat{\boldsymbol{x}}$, we infer a posterior distribution over the corresponding document embedding $p(\boldsymbol{w} \,|\, \widehat{\boldsymbol{x}})$. Parameters of such posterior distribution can be then used as a low dimensional representation of the document. Note that such distribution also encodes the inferred uncertainty about such representation.

Using Bayes' rule, the posterior distribution of document embedding $\boldsymbol{w}$ is written as:[1]

$$p(\boldsymbol{w}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{x}|\boldsymbol{w})p(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}}. \tag{5}$$

In numerator of (5), $p(\boldsymbol{w})$ represents the prior distribution of document embeddings (1), and $p(\boldsymbol{x}|\boldsymbol{w})$ represents the likelihood of observed data. According to our generative process, we assume that every document $\boldsymbol{x}$ is a sample from $\mathrm{Multinomial}$ distribution (4); hence the $\log$-likelihood is computed as follows:

$$\log p(\boldsymbol{x}|\boldsymbol{w}) = \sum_{i=1}^{V} x_i \,\log \theta_i, \tag{6}$$

$$= \sum_{i=1}^{V} x_i \log \left(\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}\}} \right), \tag{7}$$

$$= \sum_{i=1}^{V} x_i \Bigg[(m_i + \boldsymbol{t}_i \boldsymbol{w}) - \log \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}\} \right) \Bigg], \tag{8}$$

where $\boldsymbol{t}_i$ represents a row in matrix $\boldsymbol{T}$. The problem arises while computing the denominator in (5). It involves solving the integral over a product of likelihood term containing the $\mathrm{softmax}$ function and $\mathrm{Gaussian}$ distribution (prior). There exists no analytical form for this integral. This intractability is a generic problem that arises while performing Bayesian inference for mixed-logit models [22], [27], multi-class logistic regression or any other model where the likelihood $p(\boldsymbol{x}|\boldsymbol{w})$ and prior $p(\boldsymbol{w})$ are not conjugate to each other [16]. In such cases, one can resort to variational inference and find an approximation to the posterior distribution $p(\boldsymbol{w}|\boldsymbol{x})$. This approximation to the true posterior

[1]For clarity, explicit conditioning on $\boldsymbol{T}$ and $\boldsymbol{m}$ is omitted in the subsequent equations.

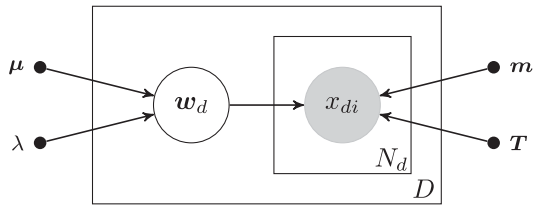


Fig. 1. Graphical model for Bayesian SMM. The arrows indicate the dependency. The rectangular plate with a symbol on the lower right corner denotes the number of repetitions of variables inside the plate. The shaded x_{di} represents the observed variable (word count), $\boldsymbol{w}_d$ represents the document-specific latent variable with hyperparameters $\boldsymbol{\mu}$ and λ. The variables $\boldsymbol{m}$ and $\boldsymbol{T}$ represent model parameters.

is referred as variational distribution $q(\boldsymbol{w})$ and is obtained by minimizing the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(q\,||\,p)$ from the approximate to the true posterior. But, computing the $D_{\mathrm{KL}}(q\,||\,p)$ also requires the functional form of true posterior $p(\boldsymbol{w}\,|\,\boldsymbol{x})$, which is intractable. Hence, we take an alternative approach to minimize the KL divergence. We express the log marginal (evidence) of the data as:

$$\log p(\boldsymbol{x}) = \mathbb{E}_q[\log p(\boldsymbol{x},\boldsymbol{w})] + \mathrm{H}[q] \; + D_{\mathrm{KL}}(q\,||\,p), \quad (9)$$

$$= \mathcal{L}(q) + D_{\mathrm{KL}}(q\,||\,p). \quad (10)$$

Here $\mathrm{H}[q]$ represents the entropy of $q(\boldsymbol{w})$. Given the data $\boldsymbol{x}$, $\log p(\boldsymbol{x})$ is a constant with respect to $\boldsymbol{w}$, and $D_{\mathrm{KL}}(q\,||\,p)$ can be minimized by maximizing $\mathcal{L}(q)$, which is known as *Evidence Lower BOund* (ELBO) for a document. This is the standard formulation of variational Bayes [16], where the problem of finding an approximate posterior is transformed into the optimization of the functional $\mathcal{L}(q)$.

III. Variational Bayes

In this section, using the VB framework, we derive and explain the procedure for estimating model parameters $\{\boldsymbol{m},\boldsymbol{T}\}$ and inferring the variational distribution, $q(\boldsymbol{w})$. Before proceeding, we note that our model assumes that all documents and the corresponding document embeddings (latent variables) are independent. This independence can be seen from the graphical model in Fig. 1. Hence, we derive the inference only for one document embedding $\boldsymbol{w}$, given an observed vector of word counts $\boldsymbol{x}$.

We chose the variational distribution $q(\boldsymbol{w})$ to be Gaussian, with mean $\boldsymbol{\nu}$ and precision $\boldsymbol{\Gamma}$, i.e., $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}\,|\,\boldsymbol{\nu},\,\boldsymbol{\Gamma}^{-1})$. The functional $\mathcal{L}(q)$ now becomes:

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\boldsymbol{x},\boldsymbol{w})] \; + \mathrm{H}[q], \quad (11)$$

$$= \mathbb{E}_q[\log p(\boldsymbol{x}\;|\;\boldsymbol{w})] + \mathbb{E}_q[\log p(\boldsymbol{w})] + H[q], \quad (12)$$

$$= \underbrace{\mathbb{E}_q[\log p(\boldsymbol{x}\;|\;\boldsymbol{w})]}_{\mathsf{A}} - \underbrace{D_{\mathrm{KL}}(q\,||\,p)}_{\mathsf{B}} \quad (13)$$

The term B in (13) is the KL divergence from the variational distribution $q(\boldsymbol{w})$ to the document-independent prior (1), which can be computed analytically [28] as:

$$D_{\mathrm{KL}}(q\,||\,p) = \frac{1}{2}\left[\lambda \mathrm{tr}(\boldsymbol{\Gamma}^{-1}) + \log|\boldsymbol{\Gamma}| - \mathrm{K}\log\lambda + \lambda\,\boldsymbol{\nu}^{\mathsf{T}}\boldsymbol{\nu} - \mathrm{K}\right], \quad (14)$$

where K denotes the document embedding dimensionality. The term A from (13) is the expectation over log-likelihood of a document (8):

$$\mathbb{E}_q[\log p(\boldsymbol{x}\,|\,\boldsymbol{w})] = \sum_{i=1}^{V} x_i \Bigg[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}) \underbrace{- \mathbb{E}_q\left[\log\Big(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\boldsymbol{w}\}\Big)\right]}_{\mathcal{F}} \Bigg]. \quad (15)$$

(15) involves solving the expectation over the log-sum-exp operation (denoted by $\mathcal{F}$), which is intractable. It appears when dealing with variational inference in mixed-logit models [22], [27]. We can approximate $\mathcal{F}$ with empirical expectation using samples from $q(\boldsymbol{w})$, but $\mathcal{F}$ is a function of $q(\boldsymbol{w})$, whose parameters we are seeking by optimizing $\mathcal{L}(q)$. The corresponding gradients of $\mathcal{L}(q)$ with respect to $q(\boldsymbol{w})$ will exhibit high variance if we directly take samples from $q(\boldsymbol{w})$ for the empirical expectation [29]. To overcome this, we will re-parametrize the random variable $\boldsymbol{w}$ by introducing a differentiable function g over another random variable $\boldsymbol{\epsilon}$ [14]. If $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, then:

$$\boldsymbol{w} = g(\boldsymbol{\epsilon}) = \boldsymbol{\nu} + \boldsymbol{L}\;\boldsymbol{\epsilon}, \quad (16)$$

where $\boldsymbol{L}$ is the Cholesky factor of $\boldsymbol{\Gamma}^{-1}$. Using this re-parametrization of $\boldsymbol{w}$, we obtain the following approximation:

$$\mathcal{F} \approx \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\,g(\tilde{\boldsymbol{\epsilon}}_r)\}\right), \quad (17)$$

where R denotes the total number of samples $\tilde{\boldsymbol{\epsilon}}_r$ from $p(\boldsymbol{\epsilon})$.

Combining (14), (15) and (17), we get the approximation to $\mathcal{L}(q)$. We will introduce the document suffix d, to make the notation explicit:

$$\mathcal{L}(q_d) \approx \; - D_{\mathrm{KL}}(q_d\,||\,p) + \sum_{i=1}^{V} x_{di}\left[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}_d) - \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\,g(\tilde{\boldsymbol{\epsilon}}_{dr})\}\right)\right]. \quad (18)$$

For the entire training data $\boldsymbol{X}$, the complete ELBO will be simply the summation over all the documents, i.e., $\sum_d \mathcal{L}(q_d)$. Note that the KL divergence term in (18) is always non-negative and is independent of the observed data, hence acts as a regularization term for the embeddings.

A. Training

The variational Bayes (VB) training procedure for Bayesian SMM is stochastic because of the sampling involved in the re-parametrization trick (16). Like the standard VB approach [16], we optimize ELBO alternately with respect to $q(\boldsymbol{w})$ and

$\{\boldsymbol{m}, \boldsymbol{T}\}$. Since we do not have closed-form update equations, we perform gradient-based updates. Additionally, we regularize rows in matrix $\boldsymbol{T}$ while optimizing. Thus, the final objective function becomes:

$$\mathcal{L} = \sum_{d=1}^{D} \mathcal{L}(q_d) - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1 \,, \tag{19}$$

where we have added the term for ℓ_1 regularization of rows in matrix $\boldsymbol{T}$, with corresponding weight ω. The same regularization was previously used for non-Bayesian SMM in [20]. This can also be seen as obtaining a maximum a posteriori estimate of $\boldsymbol{T}$ with Laplace priors.

1) Parameter Initialization: The vector $\boldsymbol{m}$ is initialized to log uni-gram probabilities estimated from training data. The values in matrix $\boldsymbol{T}$ are randomly initialized from $\mathcal{N}(0, 0.001)$. The prior over latent variables $p(\boldsymbol{w})$ is set to isotropic Gaussian distribution with mean $\mathbf{0}$ and $\lambda = \{1, 10\}$. The variational distribution $\boldsymbol{q}(w)$ is initialized to $\mathcal{N}(\mathbf{0}, (0.1)\boldsymbol{I})$. Later in Section VII-A, we will show that initializing the posterior to a sharper Gaussian distribution helps to speed up the convergence.

2) Optimization: The gradient-based updates are done by ADAM optimization scheme [23]; in addition to the following tricks:

We simplified the variational distribution $q(\boldsymbol{w})$ by making its precision matrix $\boldsymbol{\Gamma}$ diagonal.[2] Further, while updating it, we used the log standard deviation parametrization, which ensures that the variance is always positive:

$$\boldsymbol{\Gamma}^{-1} = \mathrm{diag}(\exp\{2\,\boldsymbol{\varsigma}\}). \tag{20}$$

The gradients of the objective (18) w.r.t. the mean $\boldsymbol{\nu}$ is given as follows:

$$\nabla_{\boldsymbol{\nu}} \mathcal{L} = \left[\sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}} (x_i - \frac{1}{R} \sum_{r=1}^{R} \theta_{ir} \sum_{k=1}^{V} x_k) \right] - \lambda \boldsymbol{\nu}, \tag{21}$$

where

$$\theta_{ir} = \frac{\exp\{m_i + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}}. \tag{22}$$

The gradient w.r.t log standard deviation $\boldsymbol{\varsigma}$ is given as:

$$\nabla_{\boldsymbol{\varsigma}} \mathcal{L} = \mathbf{1} - \lambda \exp\{2\boldsymbol{\varsigma}\} - \sum_{k=1}^{V} x_k \frac{1}{R} \sum_{r=1}^{R} \sum_{i=1}^{V} \theta_{ir} \boldsymbol{t}_i^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}\} \odot \boldsymbol{\epsilon}_r, \tag{23}$$

where $\mathbf{1}$ represents a column vector of ones, $\odot$ denotes element-wise product, and exp is element-wise exponential operation.

The ℓ_1 regularization term makes the objective function (19) discontinuous (non-differentiable) at points where it crosses the orthant. Hence, we used sub-gradients and employed orthant-wise learning [24]. The gradient of the objective (19) w.r.t. a row $\boldsymbol{t}_i$ in matrix $\boldsymbol{T}$ is computed as follows:

[2]This is not a limitation of the model, but only a simplification.

Algorithm 1: Stochastic VB Training.

```
initialize the model and the variational parameters
repeat
    for d = 1 ... D do
        sample ε̃_dr ~ N(0, I)        r = 1 ... R
        compute L(q_d) using (18)
        compute gradient ∇_{ν_d} L using (21)
        compute gradient ∇_{ς_d} L using (23)
        update ν_d and ς_d using ADAM
    end
    compute L using (19)
    compute sub-gradients ∇̃_{t_i} L using (24) and (25)
    update rows in T using (26)
until convergence or max_iterations
```

$$\nabla_{\boldsymbol{t}_i} \mathcal{L} = -\omega \,\mathrm{sign}(\boldsymbol{t}_i) + \sum_{d=1}^{D} \left[x_{di} \boldsymbol{\nu}_d^{\mathsf{T}} - \left[\left(\sum_{k=1}^{V} x_{ki} \right) \frac{1}{R} \sum_{r=1}^{R} \theta_{dir} (\boldsymbol{\nu}_d^{\mathsf{T}} + \boldsymbol{\epsilon}_{dr}^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}^{\mathsf{T}}\}) \right] \right]. \tag{24}$$

Here, sign and exp operate element-wise. The sub-gradient $\tilde{\nabla}_{\boldsymbol{t}_i} \mathcal{L}$ is defined as:

$$\tilde{\nabla}_{t_{ik}} \mathcal{L} \triangleq \begin{cases} \nabla_{t_{ik}} \mathcal{L} + \omega, & t_{ik} = 0, \ \nabla_{t_{ik}} \mathcal{L} < -\omega \\ \nabla_{t_{ik}} \mathcal{L} - \omega, & t_{ik} = 0, \ \nabla_{t_{ik}} \mathcal{L} > \omega \\ 0, & t_{ik} = 0, \ |\nabla_{t_{ik}} \mathcal{L}| \leq \omega \\ \nabla_{t_{ik}} \mathcal{L}, & |t_{ik}| > 0 \quad . \end{cases} \tag{25}$$

Finally, the rows in matrix $\boldsymbol{T}$ are updated according to:

$$\boldsymbol{t}_i \leftarrow \mathcal{P}_O(\boldsymbol{t}_i + \boldsymbol{d}_i) \tag{26}$$

where $\boldsymbol{d}_i$ is the step in ascent direction:

$$\boldsymbol{d}_i = \eta \mathrm{diag}(\sqrt{\boldsymbol{s}_{\mathrm{i}}} + \epsilon)^{-1} \boldsymbol{f}_{\mathrm{i}}. \tag{27}$$

Here, η is the learning rate, $\boldsymbol{f}_i$ and $\boldsymbol{s}_i$ represent bias-corrected first and second moments (as required by ADAM) of sub-gradient $\tilde{\nabla}_{\boldsymbol{t}_i} \mathcal{L}$ respectively. $\mathcal{P}_O$ represents orthant projection, which ensures that the update step does not cross the point of non-differentiability. It is defined as:

$$\mathcal{P}_O(\boldsymbol{t}_i + \boldsymbol{d}_i) \triangleq \begin{cases} 0 & \text{if } t_{ik}(t_{ik} + d_{ik}) < 0, \\ t_{ik} + d_{ik} & \text{otherwise.} \end{cases} \tag{28}$$

The orthant projection introduces explicit zeros in the estimated $\boldsymbol{T}$ matrix and results in a sparse solution. Unlike in [20], we do not require to apply the sign projection, because both the gradient $\tilde{\nabla}_{\boldsymbol{t}_i} \mathcal{L}$ and step $\boldsymbol{d}$ have same sign (point to the same orthant). The stochastic VB training is outlined in Algorithm 1.

B. Inferring Embeddings for New Documents

After obtaining the model parameters from VB training, we can infer (extract) the posterior distribution of document embedding $q(\boldsymbol{w})$ for any given document $\boldsymbol{x}$. This is done by iteratively updating the parameters of $q(\boldsymbol{w})$ that maximize $\mathcal{L}(q)$ from (18). These updates are performed by following the same ADAM optimization scheme as in training.

Note that the embeddings are extracted by maximizing the ELBO, which does not involve any supervision (topic labels). These embeddings which are in the form of posterior distributions will be used as input features for training topic ID classifiers. Alternatively, one can use only the mean of the posterior distributions as point estimates of document embeddings.

IV. Gaussian Classifier With Uncertainty

In this section, we will present a generative Gaussian classifier that exploits the uncertainty in posterior distributions of document embedding. Moreover, it also exploits the same uncertainty while computing the posterior probability of class labels. The proposed classifier is called Gaussian linear classifier with uncertainty (GLCU) and is inspired by [30], [31]. It can be seen as an extension to the simple Gaussian linear classifier (GLC) [16].

Let $\ell = 1 \dots L$ denote class labels, $d = 1 \dots D$ represent document indices, and $\boldsymbol{h}_d$ represent the class label of document d in one-hot encoding.

GLC assumes that every class is Gaussian distributed with a specific mean $\boldsymbol{\mu}_\ell$, and a shared precision matrix $\boldsymbol{D}$. Let $\boldsymbol{M}$ denote a matrix of class means, with $\boldsymbol{\mu}_\ell$ representing a column. GLC is described by the following model:

$$\boldsymbol{w}_d = \boldsymbol{\mu}_d + \boldsymbol{\varepsilon}_d, \tag{29}$$

where $\boldsymbol{\mu}_d = \boldsymbol{M}\boldsymbol{h}_d$, $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \boldsymbol{0}, \boldsymbol{D}^{-1})$ and $\boldsymbol{w}_d$ represent embedding for document d. GLC can be trained by estimating the parameters $\Theta = \{\boldsymbol{M}, \boldsymbol{D}\}$ that maximize the class conditional likelihood of all training examples. For a single training example, the likelihood is computed as:

$$p(\boldsymbol{w}_d \,|\, \boldsymbol{h}_d, \Theta) = \mathcal{N}(\boldsymbol{w}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{D}^{-1}). \tag{30}$$

In our case, however, the training examples come in the form of posterior distributions, $q(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \,|\, \boldsymbol{\nu}_d, \boldsymbol{\Gamma}_d^{-1})$ as extracted using our Bayesian SMM. In such case, the proper ML training procedure should maximize the expected class-conditional likelihood, with the expectation over $\boldsymbol{w}_d$ calculated for each training example with respect to its posterior distribution $q(\boldsymbol{w}_d)$, i.e., $\mathbb{E}_q[\mathcal{N}(\boldsymbol{w}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{D}^{-1})]$.

However, it is more convenient to introduce an equivalent model, where the observations are the means $\boldsymbol{\nu}_d$ of the posteriors $q(\boldsymbol{w}_d)$ and the uncertainty encoded in $\boldsymbol{\Gamma}_d^{-1}$ is introduced into the model through the latent variable $\boldsymbol{y}_d$ as:

$$\boldsymbol{\nu}_d = \boldsymbol{\mu}_d + \boldsymbol{y}_d + \boldsymbol{\varepsilon}_d, \tag{31}$$

where $p(\boldsymbol{y}_d) = \mathcal{N}(\boldsymbol{y}_d \,|\, \boldsymbol{0}, \boldsymbol{\Gamma}_d^{-1})$. The resulting model is called GLCU. Since the random variables $\boldsymbol{y}_d$ and $\boldsymbol{\epsilon}_d$ are Gaussian-distributed, the resulting class conditional likelihood is obtained using the convolution of two Gaussians [16]:

$$p(\boldsymbol{\nu}_d \,|\, \boldsymbol{h}_d, \Theta) = \mathcal{N}(\boldsymbol{\nu}_d \,|\, \boldsymbol{\mu}_d,\, \boldsymbol{\Gamma}_d^{-1} + \boldsymbol{D}^{-1}). \tag{32}$$

The model parameters for both GLC and GLCU have the same interpretation, i.e., each class is Gaussian distributed with specific mean and a common precision matrix. The difference lies in the evaluation of the likelihood function (30) vs (32).

GLCU can be trained by estimating its parameters Θ, that maximize the class conditional likelihood of training data (32). This can be done efficiently by using the following EM algorithm.

A. EM Algorithm

In the E-step, we calculate the posterior distribution of latent variables:

$$\begin{aligned} p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d, \boldsymbol{h}_d, \Theta) &\propto p(\boldsymbol{\nu}_d \,|\, \boldsymbol{y}_d, \boldsymbol{h}_d\Theta)\; p(\boldsymbol{y}_d) \\ &\propto \mathcal{N}(\boldsymbol{y}_d \,|\, \boldsymbol{u}_d, \boldsymbol{V}_d^{-1}), \end{aligned} \tag{33}$$

where

$$\boldsymbol{V}_d = \boldsymbol{D} + \boldsymbol{\Gamma}_d, \tag{34}$$

$$\boldsymbol{u}_d = [\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{\Gamma}_d]^{-1}(\boldsymbol{\nu}_d - \boldsymbol{\mu}_d). \tag{35}$$

In the M-step, we maximize the auxiliary function $\mathcal{Q}$ with respect to the model parameters Θ. This auxiliary function is the expectation of log joint-probability with respect to $p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d)$:

$$\mathcal{Q} = \mathbb{E}_p \left[\sum_{d=1}^{D} \log p(\boldsymbol{\nu}_d, \boldsymbol{y}_d \mid \Theta) \right] \tag{36}$$

$$\begin{aligned} = \frac{-D}{2} \log |\boldsymbol{D}| - \frac{1}{2} \Bigg[\sum_{d=1}^{D} & (\mathrm{tr}(\boldsymbol{D}\boldsymbol{V}_d^{-1}) + (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\mathsf{T}} \\ & \times \boldsymbol{D}\,(\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))) \Bigg] + \text{const.} \end{aligned} \tag{37}$$

Maximizing the auxiliary function $\mathcal{Q}$ w.r.t. Θ, we have:

$$\boldsymbol{\mu}_\ell := \frac{1}{|\mathcal{I}_\ell|} \sum_{d \in \mathcal{I}_\ell} (\boldsymbol{\nu}_d - \boldsymbol{u}_d) \quad \forall\, \ell = 1 \dots L \tag{38}$$

$$\boldsymbol{D}^{-1} := \frac{1}{D} \left[\sum_{d=1}^{D} (\boldsymbol{a}_d\, \boldsymbol{a}_d^{\mathsf{T}}) + \boldsymbol{V}_d^{-1} \right], \tag{39}$$

where $\boldsymbol{a}_d = \boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d)$, and $\mathcal{I}_\ell$ is the set of documents from the class ℓ. To train the GLCU model, we alternate between E-step and M-step until convergence.

B. Classification

Given the posterior distribution of a test document embedding $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{\nu}, \boldsymbol{\Gamma}^{-1})$, we compute the class conditional likelihood according to (32), and the posterior probability of a class $\mathcal{C}_k$ is obtained by applying the Bayes' rule:

$$p(\mathcal{C}_k \,|\, \boldsymbol{\nu}, \boldsymbol{\Gamma}, \Theta) = \frac{p(\boldsymbol{\nu} \mid \boldsymbol{\mu}_k, \boldsymbol{D}, \boldsymbol{\Gamma})\, p(\mathcal{C}_k)}{\sum_\ell p(\boldsymbol{\nu} \mid \boldsymbol{\mu}_\ell, \boldsymbol{D}, \boldsymbol{\Gamma})\, p(\mathcal{C}_\ell)}. \tag{40}$$

V. Related Models

In this section, we review and relate some of the popular PTMs and neural network-based document models. We begin with a brief review of LDA [21], a probabilistic generative model for the *bag-of-words* representation of documents.

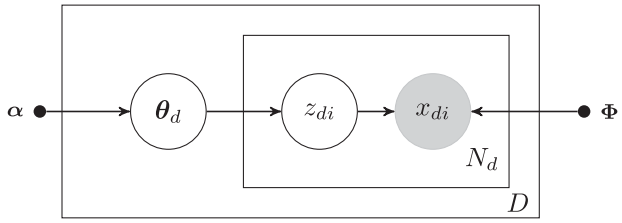


Fig. 2. Graphical model for LDA, where x_{di} is the observed variable, and $\boldsymbol{\theta}_d$, z_{di} are the latent variables.

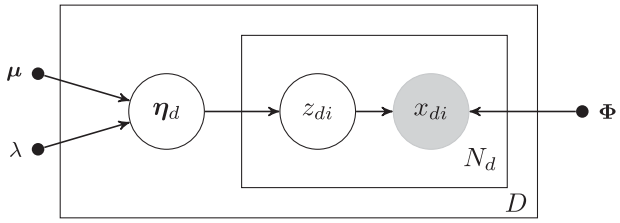


Fig. 3. Graphical model for CTM, where x_{di} is the observed variable, and $\boldsymbol{\eta}_d$, z_{di} are the latent variables.

A. Latent Dirichlet Allocation

Let $\boldsymbol{\Phi}_{K\times V}$ represent K topics, where every (row) topic $\boldsymbol{\phi}_k$ is a discrete distribution over a fixed vocabulary of size V. LDA assumes that every document d is generated by a two-step process: first, a document-specific vector (embedding) representing a discrete distribution over K topics is sampled, i.e., $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$. Then, for each word in document d, a topic indicator variable z_i is sampled: $z_i \sim \mathrm{Multi}(\boldsymbol{\theta}_d; 1)$ and the word x_i is in turn sampled from the topic-specific distribution: $x_i \sim \mathrm{Multi}(\boldsymbol{\phi}_{z_i}; 1)$.

The topic ($\boldsymbol{\phi}$) and document ($\boldsymbol{\theta}$) vectors live in $(V-1)$ and $(K-1)$ simplexes, respectively. For every word x_i in document d, there is a discrete latent variable z_i that tells which topic was responsible for generating the word. This topic-word dependency can be seen from the graphical model in Fig. 2.

During inference, the generative process is inverted to obtain posterior distribution over latent variables, $p(\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Phi})$, given the observed data $\boldsymbol{x}$ and prior belief $\boldsymbol{\alpha}$. Since the true posterior is intractable, Blei *et al.* [21] resorted to the variational inference, which finds an approximation to the true posterior as a variational distribution $q(\boldsymbol{\theta}, \boldsymbol{z})$. Further, mean-field approximation was made to make the inference tractable, i.e., $q(\boldsymbol{\theta}, \boldsymbol{z}) = q(\boldsymbol{\theta}) \prod_i q(z_i)$.

In the original model proposed by Blei *et al.* [21], the parameters $\boldsymbol{\Phi}$ were obtained using the maximum likelihood approach. The choice of $\mathrm{Dirichlet}$ distribution for $q(\boldsymbol{\theta})$ simplifies the inference process because of the $\mathrm{Dirichlet\text{-}Multinomial}$ conjugacy. However, the assumption of $\mathrm{Dirichlet}$ distribution causes limitations to the model, i.e., $q(\boldsymbol{\theta})$ cannot capture correlations between topics in each document. This was the motivation for Blei and Lafferty [22] to model documents with $\mathrm{Gaussian}$ distributions, and the resulting model is called correlated topic model (CTM).

B. Correlated Topic Model

The generative process for a document in CTM [22] is the same as in LDA, except for document vectors are now drawn from $\mathrm{Gaussian}$:

$$\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta} \mid \boldsymbol{\mu}, (\lambda \boldsymbol{I})^{-1}), \tag{41}$$

$$\boldsymbol{\theta} = \mathrm{softmax}(\boldsymbol{\eta}). \tag{42}$$

In this formulation, the document embeddings $\boldsymbol{\eta}$ are no longer in the $(K-1)$ simplex; instead they are dependent through the logistic normal. This is the same as in our proposed Bayesian SMM (1). The advantage is that the document vectors can model the correlations in topics. The topic distributions over vocabulary $\boldsymbol{\Phi}$, however, still remained discrete. In Bayesian SMM, the topic-word distributions ($\boldsymbol{T}$) are not discrete; hence it can model the correlations between words and (latent) topics [22]. The variational inference in CTM is similar to that of LDA, including the mean-field approximation, because of the discrete latent variable $\boldsymbol{z}$ (Fig. 3). An additional problem is dealing with the non-conjugacy. More specifically, it is the intractability while solving the expectation over $\log\text{-}\mathrm{sum\text{-}exp}$ function (see $\mathcal{F}$ from (15)). Blei and Lafferty [22] used Jensen's inequality to form an upper bound on $\mathcal{F}$, and this in-turn acted as a lower bound on ELBO. In our proposed Bayesian SMM, we also encountered the same problem, and we approximated $\mathcal{F}$ using the re-parametrization trick (Section III). There exist similar approximation techniques based on quasi-Monte Carlo sampling [27].

Unlike in LDA or CTM, Bayesian SMM does not require to make a mean-field approximation, because the topic-word mixture is not discrete, thus eliminating the need for discrete latent variable $\boldsymbol{z}$.

C. Subspace Multinomial Model

SMM is a log-linear model; originally proposed for modeling discrete prosodic features for the task of speaker verification [25]. Later, it was used for phonotactic language recognition [32] and eventually for topic identification and document clustering [19], [20]. A similar model was proposed by Maas *et al.* [33] for unsupervised learning of word representations. One of the significant differences among these works is the type of regularization used for matrix $\boldsymbol{T}$.

Another difference is in obtaining embeddings $\boldsymbol{w}_d$ for a given test document. Maas *et al.* [33] obtained them by projecting the vector of word counts $\boldsymbol{x}_d$ onto the matrix $\boldsymbol{T}$, i.e., $\boldsymbol{w}_d = \boldsymbol{T}\,\boldsymbol{x}_d$, whereas the authors from [19], [20] extracted the embeddings by maximizing regularized $\log$-likelihood function. However, the embeddings extracted using SMM are prone to over-fitting, especially when the observed documents are short. Our Bayesian SMM overcomes this problem by capturing the uncertainty of document embeddings in the posterior distribution. Our experimental analysis in Section VII-C illustrates the robustness of Bayesian SMM.

D. Paragraph Vector

Paragraph vector bag-of-words (PV-DBOW) [8] is also a log-linear model, which is trained stochastically to maximize the likelihood of a set of words from a given document. SMM can be seen as a special case of PV-DBOW since it maximizes the likelihood of all the words in a document.

TABLE I
DATA SPLITS FROM *FISHER* PHASE 1 CORPUS, WHERE EACH DOCUMENT REPRESENTS ONE SIDE OF THE CONVERSATION

Set	# docs.	Duration (hrs.)
ASR training	6208	553
Topic ID training	2748	244
Topic ID test	2744	226

E. Neural Network-Based Models

Neural variational document model (NVDM) is an adaptation of variational auto-encoders for document modeling [15]. The encoder models the posterior distribution of latent variables given the input, i.e., $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$, and the decoder models distribution of input data given the latent variable, i.e., $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. In NVDM, the authors used *bag-of-words* as input, while their encoder and decoders are two-layer feed-forward neural networks. The decoder part of NVDM is similar to Bayesian SMM, as both the models maximize expected $\log$-likelihood of the data, assuming $\mathrm{Multinomial}$ distribution. In simple terms, Bayesian SMM is a decoder with a single feed-forward layer. For a given test document, in NVDM, the approximate posterior distribution of latent variables is obtained directly by forward propagating through the encoder; whereas in Bayesian SMM, it is obtained by iteratively optimizing ELBO. The experiments in Section VII show that the posterior distributions obtained from Bayesian SMM represent the data better as compared to the ones obtained directly from the encoder of NVDM.

F. Sparsity in Topic Models

Sparsity is often one of the desired properties in topic models [34], [35]. Sparse coding inspired topic model was proposed in [36], where the authors have obtained sparse representations for both documents and words. ℓ_1 regularization over rows in $\boldsymbol{T}$ matrix of SMM (ℓ_1 SMM) was observed to yield better results when compared to LDA, STC and ℓ_2 regularized SMM (ℓ_2 SMM) [20]. The relation between SMM and sparse additive generative model (SAGE) [34] was explained in [19]. In [37], the authors proposed an algorithm to obtain sparse document embeddings (called sparse composite document vector (SCDV)) from pre-trained word embeddings. In our proposed Bayesian SMM, we introduce sparsity into the model parameters $\boldsymbol{T}$ by applying ℓ_1 regularization and using orthant-wise learning.

VI. EXPERIMENTS

A. Datasets

We have conducted experiments on two benchmark datasets [19], [38], [39] from speech and NLP communities. The first one is *Fisher* speech corpus,[3] which is a collection of 5850 conversational telephone speech recordings with a closed set of 40 topics. Each conversation is approximately 10 minutes long with two sides of the call and is supposedly about one topic. We considered each side of the call (recording) as an independent document, which resulted in a total of 11700 documents. Table I presents the details of data splits; they are the same as used in earlier research [19], [40], [41]. Our pre-processing involved removing punctuation and special characters, but we did not remove any stop words. Using Kaldi open-source toolkit [42], we trained a sequence-discriminative DNN-HMM automatic speech recognizer (ASR) system [43] to obtain automatic transcriptions. The ASR system resulted in 18% word-error-rate on a held-out test set. We report experimental results on both manual and automatic transcriptions. The vocabulary size while using manual transcriptions was 24854, for automatic, it was 18292, and the average document length is 830, and 856 words respectively.

The text corpus used is *20Newsgroups*,[4] which is a benchmark dataset for evaluating topic models [15], [44]–[46]. It contains 11314 training and 7532 test documents over 20 topics. Our pre-processing involved removing punctuation and words that do not occur in at least two documents, which resulted in a vocabulary of 56433 words. The average document length is 290 words.

B. Hyperparameters of Bayesian SMM

In our topic ID experiments, we observed that the embedding dimension (K) and regularization weight (ω) for rows in matrix $\boldsymbol{T}$ are the two important hyperparameters. The embedding dimension was chosen from $K = \{100, \ldots, 800\}$, and regularization weight from $\omega = \{0.0001, \ldots, 10.0\}$.

C. Proposed Topic ID Systems

Our Bayesian SMM is an unsupervised model trained iteratively by optimizing the ELBO; it does not necessarily correlate with the performance of topic ID. This is valid for SMM, NVDM or any other generative model trained without supervision. A typical way to overcome this problem is to have an early stopping mechanism (ESM), which requires to evaluate the topic ID accuracy on a held-out (or cross-validation) set at regular intervals during the training. It can then be used to stop the training earlier if needed.

Using the above-described scheme, we trained three different classifiers: (i) Gaussian linear classifier (GLC), (ii) multi-class logistic regression (LR), and, (iii) Gaussian linear classifier with uncertainty (GLCU). Note that GLC and LR cannot exploit the uncertainty in the document embeddings; and are trained using only the mean parameter $\boldsymbol{\nu}$ of the posterior distributions; whereas GLCU is trained using the full posterior distribution $q(\boldsymbol{w})$, i.e., along with the uncertainties of document embeddings as described in Section IV. GLC and GLCU do not have any hyperparameters to tune, while the ℓ_2 regularization weight for the parameters of LR was tuned using cross-validation experiments. Our code is available online.[5]

D. Baseline Topic ID Systems

1) NVDM: Since NVDM and our proposed Bayesian SMM share similarities, we chose to extract the embeddings from

[3][Online]. Available: https://catalog.ldc.upenn.edu/LDC2004S13

[4][Online]. Available: http://qwone.com/~jason/20Newsgroups/

[5][Online]. Available: https://github.com/BUTSpeechFIT/BaySMM

NVDM and use them for training linear classifiers. Given a trained NVDM model, embeddings for any test document can be extracted just by forward propagating through the encoder. Although this is computationally cheaper, one needs to decide when to stop the training, as a fully converged NVDM may not yield optimal embeddings for discriminative tasks such as topic ID. Hence, we used the same early stopping mechanism, as described in the earlier section. We used the same three classifier pipelines (LR, GLC, GLCU) as we used for Bayesian SMM. Our architecture and training scheme are similar to ones proposed in [15], i.e., two feed-forward layers with either 500 or 1000 hidden units and $\{\text{sigmoid}, \text{ReLU}, \tanh\}$ activation functions. The latent dimension was chosen from $K = \{100, \ldots, 800\}$. The hyperparameters were tuned based on cross-validation experiments.

2) SMM: Our second baseline system is non-Bayesian SMM with ℓ_1 regularization over the rows in $\boldsymbol{T}$ matrix, i.e., ℓ_1 SMM. It was trained with hyperparameters such as embedding dimension $K = \{100, \ldots, 800\}$, and regularization weight $\omega = \{0.0001, \ldots, 10.0\}$. The embeddings obtained from SMM were then used to train GLC and LR classifiers. Note that we cannot use GLCU here, because SMM yields only point-estimates of embeddings. We used the same early stopping mechanism to train the classifiers. The experimental analysis in Section VII-C shows that Bayesian SMM is more robust to over-fitting when compared to SMM and NVDM, and does not require an early stopping mechanism.

3) ULMFiT: The third baseline system is the universal language model fine-tuned for classification (ULMFiT) [9]. The pre-trained[6] model consists of 3 BiLSTM layers. Fine-tuning the model involves two steps: (a) fine-tuning LM on the target dataset and (b) training classifier (MLP layer) on the target dataset. We trained several models with various drop-out rates. More specifically, the LM was fine-tuned for 15 epochs,[7] with drop-out rates from $\{0.2, \ldots, 0.6\}$. The classifier was fine-tuned for 50 epochs with drop-out rates from $\{0.2, \ldots, 0.6\}$. A held-out development set was used to tune the hyperparameters (drop-out rates, and fine-tuning epochs).

4) TF-IDF: The fourth baseline system is a standard term frequency-inverse document frequency (TF-IDF) based document representation, followed by multi-class logistic regression (LR). Although TF-IDF is not a topic model, the classification performance of TF-IDF based systems is often close to state-of-the-art systems [19]. The hyperparameter (ℓ_2 regularization weight) of LR was selected based on 5-fold cross-validation experiments on training set.

VII. Results and Discussion

A. Convergence Rate of Bayesian SMM

We observed that the posterior distributions extracted using Bayesian SMM are always much sharper than standard Normal distribution. Hence we initialized the variational distribution to $\mathcal{N}(\mathbf{0}, (0.1)\boldsymbol{I})$ to speed up the convergence. Fig. 4 shows objective (ELBO) plotted for two different initializations of variational distribution. Here, the model was trained on *20Newsgroups* corpus, with the embedding dimension $K = 100$, regularization weight $\omega = 1.0$ and prior set to standard Normal. We can observe that the model initialized to $\mathcal{N}(\mathbf{0}, (0.1)\boldsymbol{I})$ converges faster as compared to the one initialized to standard Normal. In all the further experiments, we initialized[8] both the prior and variational distributions to $\mathcal{N}(\mathbf{0}, (0.1)\boldsymbol{I})$.

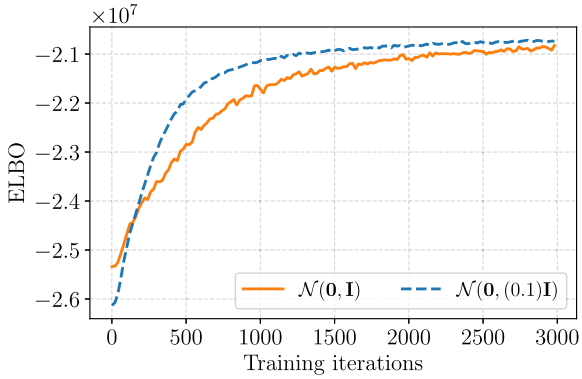


Fig. 4. Convergence of Bayesian SMM for various initializations of variational distribution. The model was trained on *20Newsgroups* corpus with $K = 100$, and $\omega = 1$.

B. Perplexity

Perplexity is inversely proportional to the log-likelihood of the data. When computed on the test data, it gives a notion of how well the model explains (fits) the test (unseen) data. Perplexity computed on test data is a standard way of evaluating language models [47], [48]. Since topic models built on bag-of-words are equivalent to unigram language models, perplexity is seen as an intrinsic measure for topic models [15], [49]. It is computed as an average of every test document according to:

$$\text{PPL}_{\text{DOC}} = \exp\left\{\frac{-1}{D}\sum_{d=1}^{D}\frac{\log p(\boldsymbol{x}_d)}{N_d}\right\}, \tag{46}$$

or for an entire test corpus according to:

$$\text{PPL}_{\text{CORPUS}} = \exp\left\{-\frac{\sum_{d=1}^{D}\log p(\boldsymbol{x}_d)}{\sum_{d=1}^{D} N_d}\right\}, \tag{47}$$

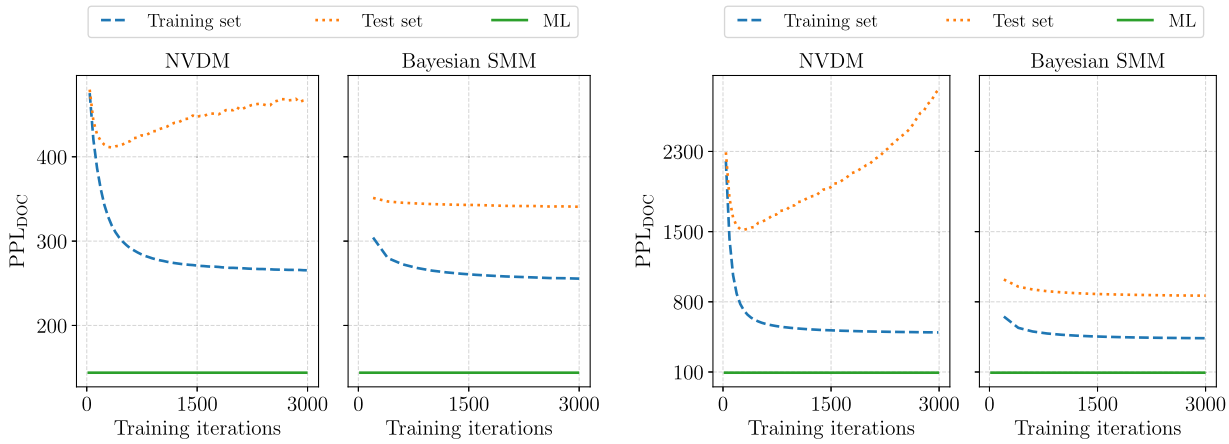
where N_d is the number of word tokens in document d.

In our case, $\log p(\boldsymbol{x})$ from (10) cannot be evaluated, because the KL divergence from variational distribution q to the true posterior p cannot be computed; as the true posterior is intractable (5). We can only compute $\mathcal{L}(q)$, which is a lower bound on $\log p(\boldsymbol{x})$. Thus, the resulting perplexity values act as upper bounds. This is true for NVDM [15] or any other model in the VB framework, where the true posterior is intractable [16]. We estimated $\mathcal{L}(q)$ from (18) using 32 samples, i.e., $R = 32$, in order to compute perplexity. We used the same number of samples for

[6][Online]. Available: https://github.com/fastai/fastai

[7]Fine-tuning LM for higher number of epochs degraded the classification performance.

[8]One can introduce hyper-priors and learn the parameters of prior distribution.



(a) PPL of *Fisher* test data. (b) PPL of *20Newsgroups* test data.

Fig. 5. Comparison of training and test data perplexities obtained using Bayesian SMM and NVDM for both *Fisher* and *20Newsgroups* datasets. The horizontal solid green line shows the test data perplexity computed using the maximum likelihood (ML) probabilities estimated on the test data. The latent (embedding) dimension was set to 200 for both the models.

TABLE II
COMPARISON OF PERPLEXITY (PPL) RESULTS ON *20NEWSGROUPS*. THE VALUES IN THE BRACKETS INDICATE RESULTS WITH A LIMITED VOCABULARY OF 2000 WORDS

Model	K	$\text{PPL}_{\text{CORPUS}}$	PPL_{DOC}
NVDM	50	1287 (769)	1421 (820)
NVDM	200	1387 (852)	1519 (870)
Bayesian SMM	50	**1043 (629)**	**1064 (639)**
Bayesian SMM	200	**882 (519)**	**851 (515)**
ML estimate	-	153 (90)	93 (42)

the baseline NVDM. We observed that perplexity values are consistent when estimated with $R \geq 16$. In [15], the authors used 20 samples.

We present the comparison of *20Newsgroups* test data perplexities obtained using Bayesian SMM and NVDM in Table II. It shows the perplexities of *20Newsgroups* corpus under full and a limited vocabulary of 2000 words [15]. We also show the perplexity computed using the maximum likelihood probabilities estimated on the test data. It acts as the lower bound on the test perplexities. NVDM was shown [15] to achieve superior perplexity scores when compared to LDA, docNADE [50], Deep Auto Regressive Neural Network models [51]. To the best of our knowledge, our model achieves state-of-the-art perplexity scores on *20Newsgroups* corpus under limited and full vocabulary conditions.

In further investigation, we trained both Bayesian SMM and NVDM until convergence. At regular checkpoints during the training, we froze the model, extracted the embeddings for both training and test data, and computed the perplexities; shown in Figs. 5(a) and 5(b). We can observe that both the Bayesian SMM and NVDM fit the training data equally well (low perplexities). However, in the case of NVDM, the perplexity of test data increases after a certain number of iterations; suggesting that NVDM fails to generalize and over-fits on the training data. In the case of Bayesian SMM, the perplexity of the test data decreases and remains stable, illustrating the robustness of our model.

C. Early Stopping Mechanism for Topic ID Systems

The embeddings extracted from a model trained purely in an unsupervised fashion does not necessarily yield optimum results when used in a supervised scenario. As discussed earlier in Section VI-C and VI-D, an early stopping mechanism (ESM) during the training of an unsupervised model (e.g., NVDM, SMM, and Bayesian SMM) is required to get optimal performance from the subsequent topic ID system. The following experiment illustrates the idea of ESM:

We trained SMM, Bayesian SMM and NVDM on *Fisher* data until convergence. At regular checkpoints during the training, we froze the model, extracted the embeddings for both training and test data. We chose GLC for SMM, GLCU for NVDM, and Bayesian SMM as topic ID classifiers. We then evaluated the topic ID accuracy on the cross-validation[9] and test sets. Fig. 6 shows the topic ID accuracy on cross-validation and test sets obtained at regular checkpoints for all the three models. The circular dot (•) represents the best cross-validation score and the corresponding test score that is obtained by employing ESM. In the case of (non-Bayesian) SMM, the test accuracy drops significantly after a certain number of iterations; suggesting the strong need of ESM. The cross-validation accuracies of NVDM and Bayesian SMM are similar and remain consistent over the iterations. However, the test accuracy of NVDM is relatively lower and decreases over the iterations. On the other hand, the test accuracy of Bayesian SMM increases and stays consistent. It shows the robustness of our proposed model, which besides does not require any ESM. In all the further topic ID experiments, we

[9]5-fold cross-validation on training set.

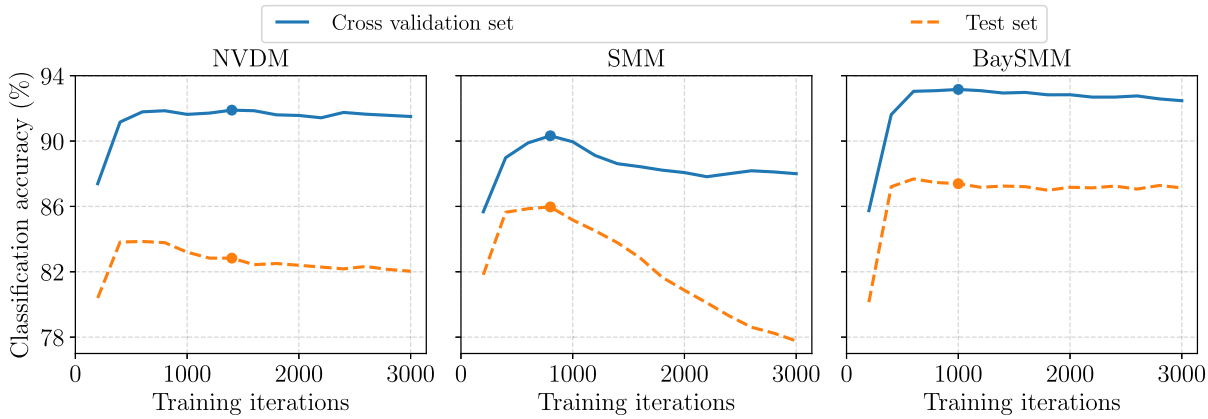


Fig. 6. Performance of topic ID systems on *Fisher* data at various checkpoints during model training. The circular dot (•) represents the best cross-validation score and the corresponding test score obtained using the early stopping mechanism (ESM). The embedding dimension was set to 100 for all the models.

TABLE III
COMPARISON OF RESULTS ON *FISHER* TEST SETS, FROM EARLIER PUBLISHED WORKS, OUR BASELINES AND PROPOSED SYSTEMS. ⋆ INDICATES A PURE DISCRIMINATIVE MODEL

Systems	Model	Classifier	Accuracy (%)	CE	Accuracy (%)	CE
			Manual transcriptions		Automatic transcriptions	
Prior works	BoW [40]	NB	87.61	-	-	-
	TF-IDF [19]	LR	86.41	-	-	-
Our Baseline	TF-IDF	LR	86.59	0.93	86.77	**0.94**
	ULMFiT ⋆	MLP	86.41	**0.50**	86.08	**0.50**
	ℓ_1 SMM	LR	86.81	0.91	87.02	1.09
	ℓ_1 SMM	GLC	85.17	1.64	85.53	1.54
	NVDM	LR	81.16	0.94	83.67	1.15
	NVDM	GLC	84.47	1.25	84.15	1.22
	NVDM	GLCU	83.96	0.93	83.01	0.97
Proposed	Bayesian SMM	LR	**89.91**	**0.89**	**88.23**	0.95
	Bayesian SMM	GLC	**89.47**	1.05	**87.23**	1.46
	Bayesian SMM	GLCU	**89.54**	**0.68**	**87.54**	**0.77**

report classification results for Bayesian SMM without ESM; while the results for SMM and NVDM are with ESM.

D. Topic ID Results

This section presents the topic ID results in terms of classification accuracy (in %) and cross-entropy (CE) on the test sets. Cross-entropy gives a notion of how confident the classifier is about its prediction. A well-calibrated classifier tends to have lower a cross-entropy.

Table III presents the classification results on *Fisher* speech corpora with manual and automatic transcriptions, where the first two rows are the results from earlier published works. Hazen [40], used discriminative vocabulary selection followed by a naïve Bayes (NB) classifier. Having a limited (small) vocabulary is the major drawback of this approach. Although we have used the same training and test splits, May *et al.* [19] had a slightly larger vocabulary than ours, and their best system is similar to our baseline TF-IDF based system. The remaining rows in Table III show our baselines and proposed systems. We can see that our proposed systems achieve consistently better accuracies; notably, GLCU, which exploits the uncertainty in document embeddings has much lower cross-entropy than its counterpart GLC. To the best of our knowledge, the proposed systems achieve the best classification results on *Fisher* corpora with the current set-up, i.e., treating each side of the conversation as an independent document. It can be observed that ULMFiT has the lowest cross-entropy among all the systems.

Table IV presents classification results on *20Newsgroups* dataset. The first three rows give the results as reported in earlier works. Pappagari *et al.* [39], proposed a CNN-based discriminative model trained to jointly optimize categorical cross-entropy loss for classification task along with binary cross-entropy for verification task. Sparse composite document vector (SCDV) [37] exploits pre-trained word embeddings to obtain sparse document embeddings, whereas neural tensor skip-gram model (NTSG) [52] extends the idea of a skip-gram model for obtaining document embeddings. The authors in (SCDV) [37] have shown superior classification results as compared to paragraph vector (PV-DM, PV-DBOW), LDA, NTSG, and other systems. The next rows in Table IV present our baselines and proposed systems. We see that the topic ID systems based on Bayesian SMM and logistic regression is better than all the other models, except for the discriminative CNN model. We can

TABLE IV
COMPARISON OF RESULTS ON *20NEWSGROUPS* FROM EARLIER PUBLISHED WORKS, OUR BASELINES AND PROPOSED SYSTEMS. ⋆ INDICATES A PURE DISCRIMINATIVE MODEL

Systems	Model	Classifier	Accuracy (%)	CE
Prior works	CNN [39] ⋆	-	**86.12**	-
	SCDV [37]	SVM	**84.60**	-
	NTSG-1 [52]	SVM	82.60	-
Our Baselines	TF-IDF	LR	84.47	**0.73**
	ULMFiT ⋆	MLP	83.06	0.89
	ℓ_1 SMM	LR	82.01	**0.75**
	ℓ_1 SMM	GLC	82.02	1.33
	NVDM	LR	79.57	0.86
	NVDM	GLC	77.60	1.65
	NVDM	GLCU	76.86	0.88
Proposed	Bayesian SMM	LR	**84.65**	**0.53**
	Bayesian SMM	GLC	83.22	1.28
	Bayesian SMM	GLCU	82.81	0.79

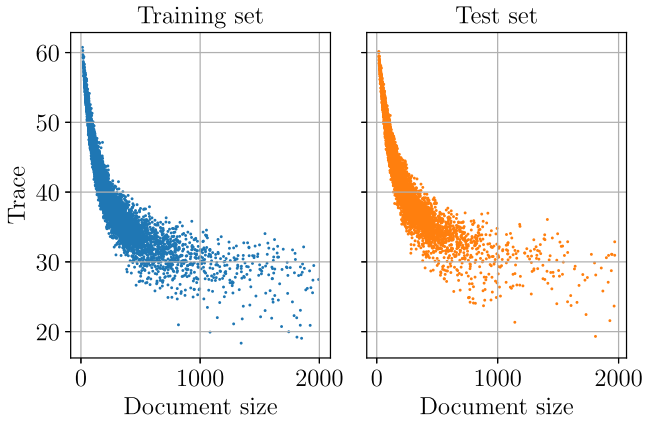


Fig. 7. Uncertainty (trace of covariance of posterior distribution) captured in the document embeddings of *20Newsgroups* dataset.

also see that all the topic ID systems based on Bayesian SMM are consistently better than variational autoencoder inspired NVDM, and (non-Bayesian) SMM.

In general, discriminative classifiers tend to perform better than generative classifiers, since discriminative classifiers model $p(y|x)$ directly, whereas generative classifiers model $p(x, y)$ to determine $p(y|x)$. However, in the presence of lower training data, generative classifiers tend to be better [53]. For instance, the cross-entropy results on *Fisher* test set (Table III) show that GLCU is better than LR. The same is not observed on *20Newsgroups* test set, where the training data is $5\times$ more when compared to *Fisher*. Moreover, Gaussian based generative classifiers are much faster to train and can be easily adapted to newer classes when compared to discriminative classifiers such as logistic regression.

The advantages of the proposed Bayesian SMM are summarized as follows: (a) the document embeddings are Gaussian distributed which enables to train simple generative classifiers like GLC, or GLCU; that can be extended to newer classes easily, (b) although the Bayesian SMM is trained in an unsupervised fashion, it does not require any early stopping mechanism to yield optimal topic ID results; document embeddings extracted from a fully converged model can be directly used for classification tasks without any fine-tuning.

E. Uncertainty in Document Embeddings

The uncertainty captured in the posterior distribution of document embeddings correlates strongly with the size of the document. The trace of the covariance matrix of the inferred posterior distributions gives us the notion of uncertainty. Fig. 7 shows an example of the uncertainty captured in the embeddings. Here, the Bayesian SMM was trained on *20Newsgroups* with an embedding dimension of 100.

VIII. CONCLUSION

We have presented a generative model for learning document representations (embeddings) and their uncertainties. Our proposed model achieved state-of-the-art perplexity results on the standard *20Newsgroups* and *Fisher* datasets. Next, we have shown that the proposed model is robust to over-fitting and unlike in SMM and NVDM, it does not require any early stopping mechanism for topic ID. We proposed an extension to simple Gaussian linear classifier that exploits the uncertainty in document embeddings and achieves better cross-entropy scores on the test data as compared to the simple GLC. Using simple linear classifiers on the obtained document embeddings, we achieved superior classification results on Fisher speech *20Newsgroups* text corpora. We also addressed a commonly encountered problem of intractability while performing variational inference in mixed-logit models by using the re-parametrization trick. This idea can be translated straightforwardly to the subspace n-gram model for learning sentence embeddings, and also for learning word embeddings along with their uncertainties. The proposed Bayesian SMM can be extended to have topic-specific priors for document embeddings, which enables to encode topic label uncertainty explicitly in the document embeddings. There exist other scoring mechanisms that exploit the uncertainty in embeddings [54], which we plan to explore in our future works.

APPENDIX A
GRADIENTS OF LOWER BOUND

The variational distribution is Gaussian with the following parameterization:

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{\nu}, \mathrm{diag}(\exp\{2\boldsymbol{\varsigma}\})). \tag{45}$$

The lower bound for a single document is:

$$\mathcal{L}_d \approx -\frac{1}{2}\Big[\lambda\,\mathrm{tr}(\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}\})) - \log|\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}\})| - K\,\log\lambda + \lambda\boldsymbol{\nu}^{\mathsf{T}}\boldsymbol{\nu} - K\Big] + \sum_{i=1}^{V} x_i\left[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}) - \frac{1}{R}\sum_{r=1}^{R}\log\left(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\, g(\boldsymbol{\epsilon}_r)\}\right)\right], \tag{46}$$

where

$$g(\boldsymbol{\epsilon}) = \boldsymbol{\nu} + \operatorname{diag}(\exp\{\boldsymbol{\varsigma}\})\tilde{\boldsymbol{\epsilon}}. \tag{47}$$

It is convenient to have the following derivatives:

$$\frac{\partial g(\boldsymbol{\epsilon})}{\partial \boldsymbol{\nu}} = \boldsymbol{I}. \tag{48}$$

$$\begin{aligned}\frac{\partial(\boldsymbol{t}_i g(\boldsymbol{\epsilon}))}{\partial \boldsymbol{\varsigma}} &= \operatorname{diag}(\boldsymbol{t}_i^{\mathsf{T}})\operatorname{diag}(\exp\{\boldsymbol{\varsigma}\})\operatorname{diag}(\tilde{\boldsymbol{\epsilon}}) \\ &= \boldsymbol{t}_i^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}}.\end{aligned} \tag{49}$$

A. Derivatives of the Parameters of Variational Distribution

Taking derivative of the objective function (46) with respect to mean parameter $\boldsymbol{\nu}$ and using (48):

$$\begin{aligned}\frac{\partial \mathcal{L}_d}{\partial \boldsymbol{\nu}} &= -\lambda\boldsymbol{\nu} + \sum_{i=1}^{V} x_i \Bigg[\boldsymbol{t}_i^{\mathsf{T}} - \frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{V}\boldsymbol{t}_k^{\mathsf{T}}\boldsymbol{I} \\ &\qquad \times \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k\, g(\boldsymbol{\epsilon}_r)\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j\, g(\boldsymbol{\epsilon}_r)\}}}_{\theta_{kr}}\Bigg]\end{aligned} \tag{50}$$

$$= \left[\sum_{i=1}^{V} x_i \boldsymbol{t}_i^{\mathsf{T}} - \sum_{i=1}^{V}\boldsymbol{t}_i^{\mathsf{T}}\frac{1}{R}\sum_{r=1}^{R}\theta_{ir}\sum_{k=1}^{V}x_k\right] - \lambda\boldsymbol{\nu} \tag{51}$$

$$\boxed{\nabla_{\boldsymbol{\nu}}\mathcal{L} = \left[\sum_{i=1}^{V}\boldsymbol{t}_i^{\mathsf{T}}\left(x_i - \frac{1}{R}\sum_{r=1}^{R}\theta_{ir}\sum_{k=1}^{V}x_k\right)\right] - \lambda\boldsymbol{\nu}.} \tag{52}$$

Taking the derivative of the objective function (46) with respect to $\boldsymbol{\varsigma}$ and using (49):

$$\begin{aligned}\frac{\partial \mathcal{L}_d}{\partial \boldsymbol{\varsigma}} =& -\frac{1}{2}\left[2\lambda\exp\{2\boldsymbol{\varsigma}\} - 2\boldsymbol{I}\right] \\ &+ \sum_{i=1}^{V} x_i\left[-\frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{V}\boldsymbol{t}_k^{\mathsf{T}}\boldsymbol{\epsilon}_r^{\mathsf{T}}\underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k g(\boldsymbol{\epsilon}_r)\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}}}_{\theta_{kr}}\right] \\ =& \mathbf{1} - \lambda\exp\{2\boldsymbol{\varsigma}\} \\ &- \left[\sum_{i=1}^{V} x_i\,\frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{V}\boldsymbol{t}_k^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}}_r^{\mathsf{T}}\theta_{kr}\right]\end{aligned} \tag{53}$$

$$\boxed{\begin{aligned}&\nabla_{\boldsymbol{\varsigma}}\mathcal{L} = \mathbf{1} - \lambda\exp\{2\boldsymbol{\varsigma}\} \\ &\quad - \left[\left(\textstyle\sum_{i=1}^{V} x_i\right)\frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{V}\theta_{kr}\boldsymbol{t}_k^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}}_r\right].\end{aligned}} \tag{54}$$

B. Derivatives of the Model Parameters

Taking the derivative of complete objective (19) with respect to a row $\boldsymbol{t}_k$ from matrix $\boldsymbol{T}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{t}_k} &= \frac{\partial}{\partial \boldsymbol{t}_k}\sum_{d=1}^{D}\sum_{i=1}^{V} x_{di}\Bigg[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}_d) \\ &\qquad - \frac{1}{R}\sum_{r=1}^{R}\log\Big(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\, g(\,\boldsymbol{\epsilon}_r\,)\}\Big)\Bigg] \\ &\quad - \omega\sum_{i=1}^{V}||\boldsymbol{t}_i||_1\end{aligned} \tag{55}$$

$$\begin{aligned}&= \sum_{d=1}^{D}\Bigg[x_{dk}\boldsymbol{\nu}_d^{\mathsf{T}} - \sum_{i=1}^{V} x_{di}\frac{1}{R}\sum_{r=1}^{R} g(\boldsymbol{\epsilon}_{dr})^{\mathsf{T}} \\ &\qquad \times \underbrace{\frac{\exp\{m_i + \boldsymbol{t}_k\, g(\boldsymbol{\epsilon}_{dr})\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_{dr})\}}}_{\theta_{dkr}}\Bigg] \\ &\quad - \omega\,\operatorname{sign}(\boldsymbol{t}_k)\end{aligned} \tag{56}$$

$$\begin{aligned}&= \sum_{d=1}^{D}\left[x_{dk}\boldsymbol{\nu}_d^{\mathsf{T}} - \sum_{i=1}^{V} x_{di}\frac{1}{R}\sum_{r=1}^{R} g(\boldsymbol{\epsilon}_{dr})^{\mathsf{T}}\theta_{dkr}\right] \\ &\quad - \omega\,\operatorname{sign}(\boldsymbol{t}_k)\end{aligned} \tag{57}$$

$$\begin{aligned}\nabla_{\boldsymbol{t}_k}\mathcal{L} = &\sum_{d=1}^{D}\left[x_{dk}\boldsymbol{\nu}_d^{\mathsf{T}} - \left[\left(\sum_{i=1}^{V} x_{di}\right)\frac{1}{R}\sum_{r=1}^{R}\theta_{dkr}g(\boldsymbol{\epsilon}_{dr})^{\mathsf{T}}\right]\right] \\ &- \omega\,\operatorname{sign}(\boldsymbol{t}_k).\end{aligned} \tag{58}$$

APPENDIX B
EM ALGORITHM FOR GLCU

E-STEP

Obtaining the posterior distribution of latent variable $p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d, \Theta)$. Using the results from [28] (p. 41, (358)):

$$\begin{aligned}&\log p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d, \boldsymbol{h}_d, \Theta) \\ &\quad= \log p(\boldsymbol{\nu}_d \mid \boldsymbol{y}_d, \boldsymbol{h}_d) + \log p(\boldsymbol{y}_d) - \log p(\boldsymbol{\nu}_d) \\ &\quad= \log\mathcal{N}(\boldsymbol{\nu}_d \mid \boldsymbol{\mu}_d + \boldsymbol{y}_d, \boldsymbol{D}^{-1}) \\ &\qquad + \log\mathcal{N}(\boldsymbol{y}_d \mid \mathbf{0}, \boldsymbol{\Gamma}_d^{-1}) + \text{const} \\ &\quad= -\frac{1}{2}(\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d))^{\mathsf{T}}\boldsymbol{D}(\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d)) \\ &\qquad - \frac{1}{2}\boldsymbol{y}_d^{\mathsf{T}}\boldsymbol{\Gamma}_d\boldsymbol{y}_d + \text{const} \\ &\quad= -\frac{1}{2}(\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\mathsf{T}}\boldsymbol{D}(\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d)) \\ &\qquad - \frac{1}{2}\boldsymbol{y}_d^{\mathsf{T}}\boldsymbol{\Gamma}_d\boldsymbol{y}_d + \text{const} \\ &\quad= \mathcal{N}(\boldsymbol{y}_d \mid \boldsymbol{u}_d, \boldsymbol{V}_d^{-1})\end{aligned}$$

where $\boldsymbol{u}_d$ is simplified as:

$$\begin{aligned}\boldsymbol{u}_d &= (\boldsymbol{D}+\boldsymbol{\Gamma}_d)^{-1}(\boldsymbol{D}(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d)+\boldsymbol{\Gamma}_d\boldsymbol{0}) \\ &= [\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{\Gamma}_d)]^{-1}(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d)\end{aligned}$$

resulting in:

$$\boldsymbol{u}_d = [\boldsymbol{I}+\boldsymbol{D}^{-1}\boldsymbol{\Gamma}_d]^{-1}(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d) \tag{59}$$

$$\boldsymbol{V}_d = \boldsymbol{D}+\boldsymbol{\Gamma}_d \tag{60}$$

M-STEP

Maximizing the auxiliary function:

$$\Theta^{\text{new}} = \underset{\Theta}{\operatorname{argmax}}\ \mathcal{Q}(\Theta,\Theta^{\text{old}}) \tag{61}$$

$$q(\boldsymbol{y}) = p(\boldsymbol{y}\mid \boldsymbol{w},\Theta^{\text{old}}). \tag{62}$$

Using the results from [28] [p. 43, (378)], the auxiliary function $\mathcal{Q}(\Theta,\Theta^{\text{old}})$ is computed as:

$$\begin{aligned}&\mathcal{Q}(\Theta,\Theta^{\text{old}}) \\ &= \mathbb{E}_q\left[\sum_{d=1}^{D}\log p(\boldsymbol{\nu}_d,\boldsymbol{y}_d)\right] \\ &= \sum_{d=1}^{D}\mathbb{E}_q[\log p(\boldsymbol{\nu}_d\mid\boldsymbol{y}_d)]+\mathbb{E}_q[\log p(\boldsymbol{y}_d)] \\ &= \sum_{d=1}^{D}\mathbb{E}_q[\log\mathcal{N}(\boldsymbol{\nu}_d\mid\boldsymbol{\mu}_d+\boldsymbol{y}_d,\boldsymbol{D}^{-1})]+\text{const} \\ &= \frac{D}{2}\log|\boldsymbol{D}| - \frac{1}{2}\sum_{d=1}^{D}\Big[\mathbb{E}_q[(\boldsymbol{\nu}_d-(\boldsymbol{\mu}_d+\boldsymbol{y}_d))^{\mathsf{T}}\boldsymbol{D} \\ &\qquad (\boldsymbol{\nu}_d-(\boldsymbol{\mu}_d+\boldsymbol{y}_d))]\Big]+\text{const} \\ &= \frac{D}{2}\log|\boldsymbol{D}| - \frac{1}{2}\sum_{d=1}^{D}\big[\operatorname{tr}(\boldsymbol{D}\boldsymbol{V}_d^{-1}) \\ &\quad +(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))^{\mathsf{T}}\boldsymbol{D}(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))\big]\end{aligned}$$

Maximizing the auxiliary function $\mathcal{Q}$ with respect to model parameters $\Theta=\{\boldsymbol{M},\boldsymbol{D}\}$

Taking the derivative with respect to each column $\boldsymbol{\mu}_\ell$ in $\boldsymbol{M}$ and equating it to zero:

$$\begin{aligned}\frac{\partial\mathcal{Q}}{\partial\boldsymbol{\mu}_\ell} &= -\frac{1}{2}\frac{\partial}{\partial\boldsymbol{\mu}_\ell}\sum_{d\in\mathcal{I}_\ell}\big[(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_\ell))^{\mathsf{T}}\boldsymbol{D}(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_l))\big] \\ &= -\frac{1}{2}\sum_{d\in\mathcal{I}_\ell}2\boldsymbol{D}\ (\boldsymbol{\mu}_\ell-(\boldsymbol{\nu}_d-\boldsymbol{u}_d)) \\ &= -\boldsymbol{D}\left(\sum_{d\in\mathcal{I}_\ell}\boldsymbol{\mu}_\ell-\sum_{d\in\mathcal{I}_\ell}(\boldsymbol{\nu}_d-\boldsymbol{u}_d)\right)\end{aligned} \tag{63}$$

$$\boldsymbol{\mu}_\ell = \frac{1}{|\mathcal{I}_\ell|}\sum_{d\in\mathcal{I}_\ell}(\boldsymbol{\nu}_d-\boldsymbol{u}_d) \tag{64}$$

Taking the derivative with respect to shared precision matrix $\boldsymbol{D}$ and equating it to zero:

$$\begin{aligned}\frac{\partial\mathcal{Q}}{\partial\boldsymbol{D}} &= \frac{D}{2}\boldsymbol{D}^{-1}-\frac{1}{2}\left(\sum_{d=1}^{D}\boldsymbol{V}_d^{-1}\right)^{\mathsf{T}} \\ &\quad -\frac{1}{2}\left(\sum_{d=1}^{D}(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))^{\mathsf{T}}\right)^{\mathsf{T}}\end{aligned} \tag{65}$$

$$\begin{aligned}\boldsymbol{D}^{-1} = \frac{1}{D}\Bigg[&\sum_{d=1}^{D}\boldsymbol{V}_d^{-1} \\ &+\sum_{d=1}^{D}(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))(\boldsymbol{u}_d-(\boldsymbol{\nu}_d-\boldsymbol{\mu}_d))^{\mathsf{T}}\Bigg]\end{aligned} \tag{66}$$

REFERENCES

[1] X. Wei and W. B. Croft, "LDA-based document models for ad-hoc retrieval," in *Proc. 29th Annu. Int. ACM SIGIR*, Aug. 2006, pp. 178–185.

[2] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. IEEE SLT Workshop*, Dec. 2012, pp. 234–239.

[3] J. Wintrode and S. Khudanpur, "Limited resource term detection for effective topic identification of speech," in *Proc. IEEE ICASSP*, May 2014, pp. 7118–7122.

[4] X. Chen *et al.*, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *Proc. Interspeech*. ISCA, Sep. 2015, pp. 3511–3515.

[5] K. Beneš, S. Kesiraju, and L. Burget, "i-Vectors in language modeling: An efficient way of domain adaptation for feed-forward models," in *Proc. Interspeech*. ISCA, 2018, pp. 3383–3387.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. NIPS*, Dec. 2013, pp. 3111–3119.

[7] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. EMNLP, ACL*, Oct. 2014, pp. 1532–1543.

[8] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. ICML*, Jun. 2014, pp. 1188–1196.

[9] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meet. ACL*. Melbourne, Australia: ACL, Jul. 2018, pp. 328–339.

[10] M. Peters *et al.*, "Deep contextualized word representations," in *Proc. NAACL: HLT*. ACL, Jun. 2018, pp. 2227–2237.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. NAACL: HLT, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Assoc. for Comput. Linguist., Jun. 2019, pp. 4171–4186.

[12] C. Bishop, "Latent variable models," in *Proc. Learn. Graph. Models*. MIT Press, Jan. 1999, pp. 371–403.

[13] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[14] D. P. Kingma and M. Welling, "Auto-Encoding variational Bayes," in *Proc. 2nd ICLR*, 2014.

[15] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proc. 33rd ICML*, ser. ICML'16. JMLR.org, 2016, pp. 1727–1736.

[16] C. M. Bishop, *Pattern Recognit. Mach. Learn. (Inform. Sci. Statist.)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[17] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic back propagation and approximate inference in deep generative models," in *Proc. 31st ICML*, ser. Proc. of Mach. Learning Res., vol. 32. Bejing, China: PMLR, 22–24 Jun. 2014, pp. 1278–1286.

[18] M. Soufifar, L. Burget, O. Plchot, S. Cumani, and J. Cernocký, "Regularized subspace n-gram model for phonotactic ivector extraction," in *Proc. Interspeech*. ISCA, Aug. 2013, pp. 74–78.

[19] C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. V. Durme, "Topic identification and discovery on text and speech," in *Proc. Conf. EMNLP*, Sep. 2015, pp. 2377–2387.

[20] S. Kesiraju, L. Burget, I. Szöke, and J. Černocký, "Learning document representations using subspace multinomial model," in *Proc. Interspeech*. ISCA, Sep. 2016, pp. 700–704.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[22] D. M. Blei and J. D. Lafferty, "Correlated topic models," in *Adv. Neural Inform. Process. Syst. NIPS*, Dec. 2005, pp. 147–154.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, May 2015.

[24] G. Andrew and J. Gao, "Scalable Training of L1-Regularized log-linear models," in *Proc. 24th ICML*. New York, USA: ACM, 2007, pp. 33–40.

[25] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Černocký, "Prosodic speaker verification using subspace multinomial models with intersession compensation," in *Proc. Interspeech*. ISCA, Sep. 2010, pp. 1061–1064.

[26] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech & Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[27] N. Depraetere and M. Vandebroek, "A comparison of variational approximations for fast inference in mixed logit models," *Comput. Statist.*, vol. 32, no. 1, pp. 93–125, 2017.

[28] K. B. Petersen and M. S. Pedersen, "The Matrix Cookbook," Nov. 2012.

[29] J. Paisley, D. M. Blei, and M. I. Jordan, "Variational Bayesian inference with stochastic search," in *Proc. 29th Int. Conf. Mach. Learn.*, ser. ICML12. Madison, WI, USA: Omnipress, 2012, pp. 1363–1370.

[30] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7649–7653.

[31] S. Cumani, O. Plchot, and R. Fér, "Exploiting i-vector posterior covariances for short-duration language recognition," in *Proc. Interspeech*, no. 9. ISCA, 2015, pp. 1002–1006.

[32] M. Soufifar *et al.*, "iVector approach to phonotactic language recognition," in *Proc. Interspeech*. ISCA, Aug. 2011, pp. 2913–2916.

[33] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meet. ACL: Human Lang. Technol.*, Jun. 2011, pp. 142–150.

[34] J. Eisenstein, A. Ahmed, and E. P. Xing, "Sparse additive generative models of text," in *Proc. 28th ICML*. USA: Omnipress, 2011, pp. 1041–1048.

[35] M. V. S. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete latent variable decomposition of counts data," in *Proc. Neural Inf. Process. Syst.*, Dec. 2007, pp. 1313–1320.

[36] J. Zhu and E. P. Xing, "Sparse topical coding," in *Proc. 27th Conf. UAI*, Jul. 2011, pp. 831–838.

[37] D. Mekala, V. Gupta, B. Paranjape, and H. Karnick, "Scdv: Sparse composite document vectors using soft clustering over distributional representations," in *Proc. Conf. Empir. Methods Natural Language Process.* Copenhagen, Denmark: ACL, Sep. 2017, pp. 659–669.

[38] S. Kesiraju *et al.*, "Topic identification of spoken documents using unsupervised acoustic unit discovery," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LA, USA, 2017, pp. 5745–5749.

[39] R. Pappagari, J. Villalba, and N. Dehak, "Joint verification-identification in end-to-end multi-scale CNN framework for topic identification," in *Proc. IEEE ICASSP*, Apr. 2018, pp. 6199–6203.

[40] T. J. Hazen, F. Richardson, and A. Margolis, "Topic identification from audio recordings using word and phone recognition lattices," in *Proc. IEEE Workshop ASRU*, Dec. 2007, pp. 659–664.

[41] T. J. Hazen, "MCE training techniques for topic identification of spoken audio documents," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2451–2460, Nov. 2011.

[42] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop ASRU*. IEEE Signal Processing Society, Dec. 2011.

[43] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*. ISCA, Aug. 2013, pp. 2345–2349.

[44] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *Proc. 34th Int. Conf. Mach. Learn. - Volume 70*, ser. ICML17. JMLR.org, 2017, pp. 2410–2419.

[45] N. Srivastava, R. Salakhutdinov, and G. E. Hinton, "Modeling documents with deep Boltzmann machines," in *UAI*, Aug. 2013.

[46] A. Srivastava and C. A. Sutton, "Autoencoding variational inference for topic models," in *Proc. 5th Int. Conf. Learn. Representations, Toulon, France*, Apr. 24-26, 2017, *Conf. Track Proc.* OpenReview.net, 2017.

[47] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 1137–1155, Mar. 2003.

[48] D. Jurafsky and J. H. Martin, *Speech and Language Process. (2nd Edition)*. USA: Prentice-Hall, Inc., 2009.

[49] N. Srivastava, R. Salakhutdinov, and G. Hinton, "Modeling documents with a deep Boltzmann machine," in *Proc. Twenty-Ninth Conf. UAI*, ser. UAI'13. Arlington, Virginia, United States: AUAI Press, 2013, pp. 616–624.

[50] H. Larochelle and S. Lauly, "A neural autoregressive topic model," in *Proc. Adv. NIPS*, Dec. 2012, pp. 2717–2725.

[51] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *Proc. 31th ICML*, Jun. 2014, pp. 1791–1799.

[52] P. Liu, X. Qiu, and X. Huang, "Learning context-sensitive word embeddings with neural tensor skip-gram model," in *Proc. 24th Int. Conf. Artif. Intell.*, ser. IJCAI'15. AAAI Press, 2015, pp. 1284–1290.

[53] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Adv. Neural Inform. Process. Syst. 14*. MIT Press, 2002, pp. 841–848.

[54] N. Brümmer, A. Silnova, L. Burget, and T. Stafylakis, "Gaussian meta-embeddings for efficient scoring of a heavy-tailed PLDA model," in *Proc. Odyssey the Speaker Lang. Recognit. Workshop*, 2018, pp. 349–356.