Learning document embeddings along with their uncertainties

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

Abstract—Majority of the text modelling 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 data better as compared to variational auto-encoder based document model. Our topic identification experiments on speech (Fisher) and text (20Newsgroups) corpora show that the proposed Bayesian SMM is robust to over-fitting on unseen test data. The topic ID results show that the proposed model is significantly better than variational auto-encoder based methods and achieve similar results when compared to fully supervised discriminative models.

Index Terms—Bayesian methods, embeddings, topic identification

I. INTRODUCTION

Learning word and document embeddings have proven to be useful in 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 modelling, and (ii) word prediction. The former methods are primarily built on top of 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].

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 learn the model parameters, and often, also the estimates of latent variables. In probabilistic topic models (PTMs) [13] the latent variables are attributed to topics, and the generative process assumes that every topic is generated from a distribution over words in the vocabulary and documents are generated from distribution of (latent) topics. Recent works showed that auto-encoders can also be seen as generative models for images and text [14], [15]. Having a generative model allows us to incorporate prior information about the latent variables, and with the help of variational Bayes (VB) techniques [16], one can infer posterior distribution over the latent variables, instead of just point estimates. The posterior distribution captures uncertainty of the latent variable estimates while trying to explain (fit) the observed data and our prior belief. In the context of text modelling, these latent variables are seen as embeddings.

In this paper, we present Bayesian subspace multinomial model (Bayesian SMM) as a generative model for bag-of-words representation of documents. We show that our model can learn to represent each document in the form of 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 [17], that can model n-gram distribution of words from 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 [18], [19]. 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 in scenarios with 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 probabilistic topic models and related algorithms for learning representations, it is important to draw precise relations between the presented model and prior research. 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 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.
correlated topic model (CTM) [21], 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 [22], [23] 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 the proposed classifier, we illustrate robustness of the model to over-fitting and at the same time achieving superior classification results when compared to SMM and NVDM.

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 III-C. Section V presents the complete VB training procedure and algorithm is presented in Section V. Experimental details are given in Section VI. Experimental details are given in Section V. Finally, we conclude and discuss directions for future research in Section VII.

II. BAYESIAN SUBSPACE MULTINOMIAL MODEL

Our generative probabilistic model assumes that the training data (bag-of-words) were generated as follows:

For each document, a $K$-dimensional latent vector $\mathbf{w}$ is generated from isotropic Gaussian prior with mean $\mathbf{0}$ and precision $\lambda$:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \text{diag}((\lambda \mathbf{I})^{-1}))$$  \hspace{1cm} (1)

The latent vector $\mathbf{w}$ 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 is calculated as:

$$\mathbf{\theta} = \text{softmax}(\mathbf{m} + \mathbf{T} \mathbf{w}),$$  \hspace{1cm} (2)

where $\{\mathbf{m}, \mathbf{T}\}$ are parameters of the model. Vector $\mathbf{m}$ known as universal background model represents log uni-gram probabilities of words. $\mathbf{T}$ known as total variability matrix [23], [24] is a low-rank matrix defining subspace of document-specific distributions.

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

$$\mathbf{x} \sim \text{Multi}(\mathbf{\theta}; N),$$  \hspace{1cm} (3)

where $N$ is the number of words in the document.

The above described generative process fully defines our Bayesian model, which we will now use to address the following problems: given training data $\mathbf{X}$, we estimate model parameters $\{\mathbf{m}, \mathbf{T}\}$ and, for any given document $\tilde{x}$, we infer posterior distribution over corresponding document embedding $p(\mathbf{w} | \tilde{x}).$ 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 $\mathbf{w}$ is written as:

$$p(\mathbf{w} | \mathbf{x}) = \frac{p(\mathbf{x} | \mathbf{w}) p(\mathbf{w})}{\int p(\mathbf{x} | \mathbf{w}) p(\mathbf{w}) d\mathbf{w}}.$$  \hspace{1cm} (4)

In numerator of Eq. (4), $p(\mathbf{x} | \mathbf{w})$ represents prior distribution of document embeddings, which is according to Eq. (1) and $p(\mathbf{x} | \mathbf{w})$ represents the likelihood of observed data. According to our generative process, we assume that every document $\mathbf{x}$ is a sample from Multinomial distribution (Eq. (3)), and the log-likelihood is given as follows:

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

$$= \sum_{i=1}^{V} x_i \left( \frac{\exp\left( m_i + t_i \mathbf{w}\right)}{\sum_{j=1}^{V} \exp\left( m_j + t_j \mathbf{w}\right)} \right),$$

$$= \sum_{i=1}^{V} x_i \left( \frac{m_i + t_i \mathbf{w}}{\sum_{j=1}^{V} \exp\left( m_j + t_j \mathbf{w}\right)} \right).$$  \hspace{1cm} (7)

where $t_i$ represents a row in matrix $\mathbf{T}$. The problem arises while computing the denominator in Eq. (4). It involves solving the integral over a product of likelihood term containing the softmax function and Gaussian distribution (prior). There exists no analytical form for this integral. This is a generic problem that arises while performing Bayesian inference for mixed-logit models, multi-class logistic regression or any other model where likelihood function and prior are not conjugate to each other [25]. In such cases, one can resort to variational techniques, both variational inference and approximate posterior inference. In what follows we describe approximate posterior inference, using variational lower bounds for the log marginal (evidence) of the data as:

$$\log p(\mathbf{x}) = \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{w})] + \mathcal{H}[q] + D_{KL}(q || p),$$

$$= \mathcal{L}(q) + D_{KL}(q || p).$$  \hspace{1cm} (9)

Here $\mathcal{H}[q]$ represents the entropy of $q(\mathbf{w})$. Given the data $\mathbf{x}$, $\log p(\mathbf{x})$ is a constant with respect to $\mathbf{w}$, and $D_{KL}(q || p)$

1For clarity, explicit conditioning on $\mathbf{T}$ and $\mathbf{m}$ is omitted in the subsequent equations.
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 [26], where the problem of finding an approximate posterior is transformed into 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 $\{m, T\}$ and inferring the variational distribution, $q(w)$. Before proceeding, we note that our model assumes that all documents and the corresponding document embeddings (latent variables) are independent. This can be seen from the graphical model in Fig. 1. Hence, we derive the inference only for one document embedding $w$, given observed vector of word counts $x$.

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

$$\mathcal{L}(q) = \mathcal{E}_q[\log p(x, w)] + H[q],$$

$$= - D_{KL}(q || p) + \mathcal{E}_q[\log p(x | w)].$$

(10)

(11)

The term $A$ from Eq. (11) is the negative KL divergence between the variational distribution $q(w)$ and the document-independent prior $p$ from Eq. (1). This can be computed analytically [27] as:

$$D_{KL}(q || p) = \frac{1}{2} \left[ \lambda \text{tr} \left( \Gamma^{-1} \right) + \log |\Gamma| - K \log \lambda + \lambda \nu^T \nu - K \right].$$

(12)

where $K$ denotes the dimension of document embedding. The term $B$ from Eq. (11) is the expectation over log-likelihood of a document (Eq. (7)):

$$\mathcal{E}_q[\log p(x | w)] = \sum_{i=1}^{V} x_i \left[ (m_i + t_i \nu) - \mathcal{E}_q \left[ \log \sum_{j=1}^{V} \exp \{m_j + t_j \nu \} \right] \right].$$

(13)

Eq. (13) involves solving the expectation over log-sum-exp operation (denoted by $\mathcal{F}$), which is intractable. It appears when dealing with variational inference in mixed-logit models [21], [28]. We can approximate $\mathcal{F}$ with empirical expectation using samples from $q(w)$, but $\mathcal{F}$ is a function of $q(w)$, whose parameters we are seeking by optimizing $\mathcal{L}(q)$. The corresponding gradients of $\mathcal{L}(q)$ with respect to $q(w)$ will exhibit high variance if we directly take samples from $q(w)$ for the empirical expectation. To overcome this, we will re-parametrize the random variable $w$ [14]. This is done by introducing a differentiable function $g$ over another random variable $\epsilon$. If $p(\epsilon) = \mathcal{N}(0, I)$, then,

$$w = g(\epsilon) = \nu + L \epsilon,$$

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

$$\mathcal{F} \approx \frac{1}{R} \sum_{r=1}^{R} \log \left( \sum_{j=1}^{V} \exp \{m_j + t_j g(\epsilon_r)\} \right),$$

(15)

where $R$ denotes the total number of samples ($\epsilon_r$) from $p(\epsilon)$. Combining Eqs. (12), (13) and (15), we get the approximation to $\mathcal{L}(q)$. We will introduce the document suffix $d$, to make the notation explicit:

$$\mathcal{L}(q_d) \approx -D_{KL}(q_d || p) + \sum_{i=1}^{V} x_{di} \left[ (m_i + t_i \nu_d) + \frac{1}{R} \sum_{r=1}^{R} \log \left( \sum_{j=1}^{V} \exp \{m_j + t_j g(\epsilon_{dr})\} \right) \right].$$

(16)

For the entire training data $X$, the complete ELBO will be simply the summation over all the documents, i.e., $\sum_d \mathcal{L}(q_d)$.

A. Training

The variational Bayes (VB) training procedure for Bayesian SMM is stochastic because of the sampling involved in the re-parametrization trick (Eq. (14)). Like the standard VB approach [26], we optimize ELBO alternately with respect to $q(w)$ and $\{m, T\}$. Since we do not have closed form update equations, we perform gradient-based updates. Additionally, we regularize rows in matrix $T$ while optimizing. Thus, the final objective function becomes,

$$\mathcal{L} = \sum_{d=1}^{D} \mathcal{L}(q_d) - \omega \sum_{i=1}^{V} ||t_i||_1,$$

(17)

where we have added term for $\ell_1$ regularization of rows in matrix $T$, with corresponding weight $\omega$. The same regularization was previously used for non Bayesian SMM in [19]. This can also be seen as obtaining a maximum a posteriori estimate of $T$ with Laplace priors.

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

2) Optimization: The gradient-based updates are done by ADAM optimization scheme [22], in addition to the following tricks:

We simplified the variational distribution $q(w)$ by making its precision matrix $\Gamma$ diagonal. Further, while updating it, we used log standard deviation parametrization, i.e.,

$$\Gamma^{-1} = \text{diag}(\exp\{2\varsigma\}).$$

(18)

This is not a limitation, but only a simplification.
The gradients of the objective w.r.t. mean $\nu$ is given as follows:

$$
\nabla \nu = \left[ \sum_{i=1}^{V} t_i^T \left( x_i - \frac{1}{R} \sum_{r=1}^{R} \theta_{ir} \sum_{k=1}^{V} x_k \right) \right] - \lambda \nu \tag{19}
$$

where,

$$
\theta_{ir} = \frac{\exp\{m_i + t_j g(\epsilon_r)\}}{\sum_j \exp\{m_j + t_j g(\epsilon_r)\}} \tag{20}
$$

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

$$
\nabla \varsigma = 1 - \lambda \exp\{2\varsigma\} - \sum_{k=1}^{V} x_k \frac{1}{R} \sum_{r=1}^{R} \sum_{i=1}^{V} \theta_{ir} t_i^T \odot \exp\{\varsigma\} \odot \epsilon_r, \quad \tag{21}
$$

where $1$ represents a column vector of 1s, $\theta_{ir}$ is according to Eq. (20), $\odot$ denotes element-wise product, and exp is element-wise exponential operation.

The $\ell_1$ regularization term makes the objective function (Eq. (17)) discontinuous (non-differentiable) at points where it crosses the orthant. Hence, we used sub-gradients and employed orthant-wise learning [23]. The gradient of the objective w.r.t. a row $t_i$ in matrix $T$ is computed as follows:

$$
\nabla t_i = -\omega \text{sign}(t_i) + \sum_{d=1}^{D} \left[ \frac{x_{d} t_d}{\sum_{k=1}^{V} R} \right] \left( \left[ \sum_{k=1}^{V} x_{k} \right] \frac{1}{R} \sum_{r=1}^{R} \theta_{dir} \left( \nu_{d}^T + \epsilon_{dr} \odot \exp\{\varsigma\} \right) \right). \tag{22}
$$

Here, sign and exp operate element-wise. The sub-gradient $\nabla t_{ik}$ is defined as:

$$
\nabla t_{ik} \triangleq \begin{cases} 
\nabla t_{ik} + \omega, & t_{ik} = 0, \quad \nabla t_{ik} < -\omega \\
\nabla t_{ik} - \omega, & t_{ik} = 0, \quad \nabla t_{ik} > \omega \\
0, & t_{ik} = 0, \quad |\nabla t_{ik}| \leq \omega \\
|\nabla t_{ik}| > \omega
\end{cases} \tag{23}
$$

Finally, the rows in matrix $T$ are updated according to,

$$
t_i \leftarrow P_O(t_i + d_i), \tag{24}
$$

where,

$$
d_i = \eta \text{diag}(\sqrt{s_i} + \epsilon)^{-1} f_i, \tag{25}
$$

$$
P_O(t_i + d_i) \triangleq \begin{cases} 
0 & \text{if } t_{ik}(t_{ik} + d_{ik}) < 0, \\
\text{if } t_{ik}(t_{ik} + d_{ik}) > 0, \text{ otherwise}. & 
\end{cases} \tag{26}
$$

Here, $\eta$ is the learning rate, $f_i$ and $s_i$ represents bias corrected first and second moments (as required by ADAM) of sub-gradient $\nabla t_i$, respectively. $P_O$ represents orthant projection, which assures that the update step does not cross the point of non-differentiability. It introduces explicit zeros in the estimated $T$ matrix, and results in sparse solution. Unlike in [19], we do not require to apply the sign projection, because both gradient and step $d$ point to the same orthant (due to properties of ADAM). The stochastic VB training is outlined in Algorithm 1.

### Algorithm 1: Stochastic VB training

1. initialize model and variational parameters
2. repeat
   3. for $d = 1 \ldots D$ do
      4. sample $\epsilon_{dr} \sim \mathcal{N}(0, I)$, $r = 1 \ldots R$
      5. compute $\mathcal{L}(q_d)$ using Eq. (16)
      6. compute gradient $\nabla q_d$ using Eq. (19)
      7. compute gradient $\nabla \nu_d$ using Eq. (21)
      8. update $\nu_d$ and $\varsigma_d$ using ADAM
   9. end
   10. compute $\mathcal{L}$ using Eq. (17)
   11. compute sub-gradients $\nabla t_i$ using Eqs. (22) and (23)
   12. update rows in $T$ using Eq. 24
13. until convergence or max_iterations

### B. Inferring embeddings for new documents

After obtaining the model parameters from VB training, we can infer (extract) the document embedding posterior distributions $q(w)$, for any given document $x$. This is done by iteratively updating the parameters of $q(w)$ that maximize $\mathcal{L}(q)$ from Eq. (16). These updates are performed by following the same ADAM optimization scheme as in training.

Note that the embeddings are extracted by maximizing the ELBO, that does 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 document embedding posterior distributions. 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 from [29], [30]. It can be seen as an extension to the simple Gaussian linear classifier (GLC) [26].

Let $\ell = 1 \ldots L$ denote class labels, $d = 1 \ldots D$ represent document indices, and $h_d$ represent the class label of document $d$ in one-hot encoding.

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

$$
\mathbf{w}_d = \mu_d + \epsilon_d, \tag{27}
$$

where $\mu_d = M h_d$, $\mathbf{p}(\epsilon) = \mathcal{N}(\epsilon | 0, D^{-1})$ and $\mathbf{w}_d$ represent embedding for document $d$. GLC can be trained by estimating the parameters $\theta = \{M, D\}$ that maximize the class conditional likelihood of all training examples:

$$
\prod_{d=1}^{D} \mathbf{p}(\mathbf{w}_d | h_d, \theta) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d | \mu_d, D^{-1}). \tag{28}
$$
In our case, however, the training examples come in the form of posterior distributions, \( q(w_d) = \mathcal{N}(w_d | \nu_d, \Gamma_d^{-1}) \) 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 \( w_d \) calculated for each training example with respect to its posterior distribution \( q(w_d) \) i.e., \( \mathbb{E}_q[\mathcal{N}(w_d | \mu_d, D^{-1})] \).

However, it is more convenient to introduce an equivalent model, where the observations are the means \( \nu_d \) of the posteriors \( q(w_d) \) and the uncertainty encoded in \( \Gamma_d^{-1} \) is introduced into the model through latent variable \( y_d \) as,

\[
\nu_d = \mu_d + y_d + \epsilon_d, \quad (29)
\]

where, \( p(y_d) = \mathcal{N}(y_d | 0, \Gamma_d^{-1}) \). The resulting model is called GLCU. Since the random variables \( y_d \) and \( \epsilon_d \) are Gaussian-distributed, the resulting class conditional likelihood is obtained by convolution of two Gaussians \( \mathcal{N} \), i.e.,

\[
p(\nu_d | h_d, \Theta) = \mathcal{N}(\nu_d | \mu_d, \Gamma_d^{-1} + D^{-1}). \quad (30)
\]

GLCU can be trained by estimating its parameters \( \Theta \), that maximize the class conditional likelihood (Eq. (30)) of training data. 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:

\[
p(y_d | \nu_d, \Theta) \propto p(\nu_d | y_d, \Theta) p(y_d) \\
= \mathcal{N}(y_d | u_d, V_d^{-1}), \quad (31)
\]

where,

\[
V_d = D + \Gamma_d, \quad (32)
\]

\[
u_d = [I + D^{-1} \Gamma_d^{-1}(\nu_d - \mu_d)]. \quad (33)
\]

In the M-step, we maximize the auxiliary function \( Q \) with respect to model parameters \( \Theta \). It is the expectation of log joint-probability with respect to \( p(y_d | \nu_d) \), i.e.,

\[
Q = \mathbb{E}_p[\sum_{d=1}^{D} \log p(\nu_d, y_d | \Theta)] \\
= \frac{1}{2} \left[ \sum_{d=1}^{D} \left( \text{tr}(DV_d^{-1}) + (u_d - (\nu_d - \mu_d))^\top D(u_d - (\nu_d - \mu_d)) \right) \\
- N \log|D| \right] + \text{const.} \quad (35)
\]

Maximizing the auxiliary function \( Q \) w.r.t. \( \Theta \), we have:

\[
\mu_\ell := \frac{1}{|\mathcal{I}_\ell|} \sum_{d \in \mathcal{I}_\ell} (\nu_d - u_d) \quad \forall \ell = 1 \ldots L \quad (36)
\]

\[
D^{-1} := \frac{1}{D} \left[ \sum_{d=1}^{D} (a_d a_d^\top) + V_d^{-1} \right], \quad (37)
\]

where \( \mathcal{I}_\ell \) is the set of documents from class \( \ell \). To train the GLCU model, we alternate between E-step and M-step until convergence.

B. Classification

Given a test document embedding posterior distribution \( q(w) = \mathcal{N}(w | \nu, \Gamma^{-1}) \), we compute the class conditional likelihood according to Eq. (30), and the posterior probability of a class \( C_k \) is obtained by applying the Bayes’ rule:

\[
p(C_k | \nu, \Gamma, \Theta) = \frac{p(\nu | \mu_k, D, \Gamma) p(C_k)}{\sum_k p(\nu | \mu_k, D, \Gamma) p(C_k)}. \quad (38)
\]

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 \( [20] \), a probabilistic generative model for bag-of-words representation of documents.

A. Latent Dirichlet allocation

LDA assumes that every latent topic \( \phi \) is a discrete probability distribution over vocabulary of words, and every document is a discrete probability distribution over latent topics. The generative process for a document (bag-of-words) can be explained by the following two steps: First, a document-specific vector (embedding) \( \theta \) is sampled from \( \text{Dir}(\alpha) \) prior with parameter \( \alpha \). Then, for each word in the document, a latent topic \( z_i \) is sampled: \( z_i \sim \text{Multi}(\theta) \). Then, word \( x_i \) is in turn sampled from the topic specific distribution: \( x_i \sim \text{Multi}(\phi_{z_i}) \).

The topic \( \phi \) and document \( \theta \) 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 can be seen from the respective graphical model in Fig. 2.

During inference, the generative process is inverted to obtain posterior distribution over latent variables, \( p(\theta, z | x, \alpha, \Phi) \), given the observed data and prior belief. Since the true posterior is intractable, Blei \( [20] \) resorted to variational inference which finds an approximation to the true posterior as a variational distribution \( q(\theta, z) \). Further, mean-field approximation was made, to make the inference tractable, i.e., \( q(\theta, z) = q(\theta) \prod_i q(z_i) \).

In the original model proposed by Blei \( [20] \), the parameters \( \phi \) were obtained using maximum likelihood approach. The choice of Dirichlet distribution for \( q(\theta) \) simplifies the inference process because of the Dirichlet-Multinomial conjugacy. However, the assumption of Dirichlet distribution causes limitations to the model, and \( q(\theta) \) cannot capture correlations between topics in each document. This was the motivation for Blei \( [21] \) to model documents with Gaussian distributions, and the resulting model is called correlated topic model (CTM).
**B. Correlated topic model**

The generative process for a document in CTM [21] is same as in LDA, except for document vectors are now drawn from Gaussian, i.e.,

\[
p(\eta) = N(\eta | \mu, \text{diag}(\lambda)^{-1}),
\]

\[
\theta = \text{softmax}(\eta).
\]

In this formulation, the document embeddings \(\eta\) are no longer in the \((K - 1)\) simplex, rather they are dependent through the logistic normal (natural parametrization of the exponential family). This is the same as in our proposed Bayesian SMM (Eq. (1)). The advantage is that the document vectors can model the correlations in topics. The topic distributions over vocabulary \(\phi\), however, still remained Discrete. In Bayesian SMM, the topic-word distributions \((T)\) are not Discrete, hence it can model the correlations between words and (latent) topics.

The variational inference in CTM is similar to that of LDA including the mean-field approximation, because of the discrete latent variable \(z\) (Fig. 5). The additional problem is dealing with the non-conjugacy. More specifically, it is the intractability while solving the expectation over log-sum-exp function (see \(\mathcal{F}\) from Eq. (13)). Blei [21] used Jensen’s inequality to form an upper bound on \(\mathcal{F}\), and this in-turn acted as 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 [28].

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

**C. Subspace multinomial model**

SMM is a log-linear model, originally proposed for modelling discrete prosodic features for the task of speaker verification [24]. Later, it was used for phonotactic language recognition [31] and eventually for topic identification and document clustering [18], [19]. Similar model was proposed by Maas [32] for unsupervised learning of word representations. One of the major differences among these works is the type of regularization used for matrix \(T\). Another major difference is in obtaining embeddings \(w_d\) for a given test document. Maas [32] obtained them by projecting the vector of word counts \(x_d\) onto the matrix \(T\), i.e., \(w_d = Tx_d\), whereas [18], [19] extracted the embeddings by maximizing regularized log-likelihood function.

**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 generalization of PV-DBOW, as it maximizes the likelihood of all the words in a document.

**E. Neural network based models**

Neural variational document model (NVDM) is an adaptation of variational auto-encoders for document modelling [15]. The encoder models the posterior distribution of latent variables given the input, i.e., \(p_\eta(z | x)\), and the decoder models distribution of input data given the latent variable, i.e., \(p_\theta(x | z)\). 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 data, assuming 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 [33], [34]. Sparse coding inspired topic model was proposed by [35], where the authors have obtained sparse representations for both documents and words. \(\ell_1\) regularization over \(T\) for SMM (\(\ell_1\) SMM) was observed to yield better results when compared to LDA, STC and \(\ell_2\) regularized SMM (\(\ell_2\) SMM) [19]. Relation between SMM and sparse additive generative model (SAGE) [35] was explained in [18]. In [36], 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 by applying \(\ell_1\) regularization and using orthant-wise learning.

**VI. EXPERIMENTS**

**A. Datasets**

We have conducted experiments on both speech and text corpora. The speech data used is Fisher phase 1 corpus [4] 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

https://catalog.ldc.upenn.edu/LDC2004S13
earlier research [18], [37], [38]. Our preprocessing involved removing punctuation and special characters, but we did not remove any stop words. Using Kaldi open-source toolkit [39], we trained a sequence discriminative DNN-HMM automatic speech recognizer (ASR) system [40] 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 contains 11314 training and 7532 test documents over 20 topics. Our preprocessing 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. Hyper-parameters of Bayesian SMM

In our topic ID experiments, we observed that the embedding dimension (K) and regularization weight (ω) for rows in matrix T are the two important hyper-parameters. The embedding dimension was chosen from $K = \{100, \ldots, 800\}$, and regularization weight from $\omega = \{0.0001, \ldots, 10.0\}$. The hyper-parameters were tuned based on cross-validation experiments.

### C. Proposed topic ID systems

In our Bayesian SMM, the document embeddings are extracted (inferred) in an iterative fashion by optimizing the ELBO; this does not necessarily correlate with the performance of topic ID. It is valid for SMM, NVDM or any other generative model trained without supervision. A typical way to overcome this problem is to have a topic ID performance monitoring system (PMS), which evaluates the topic ID accuracy on a held-out (or cross-validation) set at regular intervals during the inference. The PMS can be used to stop the inference earlier if needed.

Using the above described scheme, we trained three different classifiers: (i) Gaussian linear classifier (GLC), and (ii) multi-class logistic regression (LR) are trained using only the mean parameter ($\mu_d$) of the posterior distributions, (iii) Gaussian linear classifier uncertainty (GLCU) is trained using the full posterior distribution, i.e., along with the uncertainties of document embeddings as described in Section [IV] GLC and GLCU does not have any hyper-parameters to tune, while the $\ell_2$ regularization weight of LR was tuned using cross-validation experiments.

### D. Baseline topic ID systems

1) **NVDM**: In the original paper [15], the authors did not use the embeddings from NVDM for document classification. Since NVDM and our proposed Bayesian SMM share similarities, we chose to extract the embeddings from 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 training, as a fully converged NVDM may not yield optimal embeddings for discriminative tasks such as topic ID. Hence, we used the same strategy of having a topic ID PMS. 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 $\{\text{sigmoid, ReLU, tanh}\}$ activation functions. The latent dimension was chosen from $K = \{100, \ldots, 800\}$. The hyper-parameters were tuned based on cross-validation experiments.

2) **SMM**: Our second baseline system is non-Bayesian SMM with $\ell_1$ regularization over the rows in $T$ matrix, i.e., $\ell_1$ SMM. It was trained with several hyper-parameters 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. The classifier training scheme is the same as we used in Bayesian SMM and NVDM, i.e., by using a performance monitoring system for topic ID. 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 a performance monitoring system.

3) **ULMFiT**: The third baseline system is universal language model fine-tuned for classification (ULMFiT) [9]. The pre-trained 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 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 all the hyper-parameters.

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 are often close to state-of-the-art systems [18]. The hyper-parameter ($\ell_2$ regularization weight) of LR was selected based on 5-fold cross-validation experiments on training set.
VII. RESULTS AND DISCUSSION

A. Convergence 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 $N(0, \text{diag}(0.1)I)$ for faster convergence. Fig. 4 shows objective (ELBO) plotted against training iterations for two different initializations of variational distribution. Here, the model was trained on manual transcriptions of Fisher corpus, with the embedding dimension $K = 100$, regularization weight $\omega = 0.1$ and prior set to standard Normal. We can observe that the model initialized to $N(0, \text{diag}(0.1)I)$ converges faster as compared to the one initialized to standard Normal.

B. Perplexity

Perplexity is an intrinsic measure for topic models [15], [41]. We computed it for the entire test data according to:

$$\text{Corpus PPL} = \exp \left\{ - \frac{\sum_{d=1}^{D} \log p(x_d)}{\sum_{d=1}^{D} N_d} \right\},$$  \hspace{1cm} (41)

where $N_d$ is the number of words in document $d$. For Bayesian SMM, $\log p(x_d)$ is approximated with a lower bound according to Eq. (16). It is also the case with NVDM. Since $\log p(x_d)$ is approximated with lower-bound, the resulting perplexity values act as upper bounds.

Figs. 5a and 5b compare the test data perplexities of Bayesian SMM and NVDM on both the datasets. The horizontal solid green line shows the perplexity computed using the maximum likelihood probabilities estimated on the test data. The latent (embedding) dimension was set to 100. The horizontal solid green line shows the perplexity computed using the maximum likelihood probabilities estimated on the test data.

C. Performance monitoring for topic ID systems

The embeddings extracted from a model trained purely in unsupervised fashion does not necessarily yield optimum results when used in a supervised scenario. As discussed earlier in Sections VI-C and VI-D, the topic ID systems are trained on document embeddings extracted from unsupervised trained Bayesian SMM for 5000 VB iterations; then the embeddings are extracted for test documents for 5000 iterations. The perplexity is computed at regular checkpoints during extraction, as shown in Figs. 5a and 5b. We can observe that Bayesian SMM fits the test data better than NVDM. Also, one can observe that NVDM tends to over-fit on the training data and hence the perplexities of test data increase from around iteration 400. It suggests that one requires to have a held-out set to monitor the perplexity values to decide when to stop the training.
models (SMM, Bayesian SMM, NVDM), and requires a performance monitoring system to achieve optimal results. To illustrate this, we show the topic ID accuracy against the iterative optimization of unsupervised models: NVDM, SMM and Bayesian SMM.

Similar to the perplexity experiments, we trained NVDM for 3000 iterations, and at regular intervals during training, we froze the model, extracted embeddings for both training and test documents. We then trained topic ID classifiers and evaluated the training data using 5-fold cross-validation, and also test data. In the case of SMM and Bayesian SMM, we trained the models for 5000 iterations; then embeddings were extracted 5000 iterations. At regular intervals during the extraction, we trained a topic ID system and evaluated the training set with 5-fold cross-validation, and also test set. The cross-validation results can be used to decide when to stop the iterative optimization scheme. Fig. 6 illustrates the above-described scheme, executed on Fisher data, with the embedding dimension set to 100 and the classifier set to GLC. We can observe that the embeddings extracted from non-Bayesian SMM are prone to over-fitting and hence, a steep fall in test accuracy. The cross-validation accuracies of NVDM and Bayesian SMM are similar and consistent over the iterations. However, the test accuracy of NVDM is much lower than that of Bayesian SMM and also decreases over iterations. On the other hand, the test accuracy of Bayesian SMM increases and stays consistent. It shows the robustness of our proposed model, which in addition, does not require any performance monitoring system for topic ID.

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 cross-entropy.

Table II 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 [37], 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 [18] had slightly larger vocabulary than ours, and their best system is similar to our baseline TF-IDF based system. The remaining rows in Table II 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 counter part, 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.

Table III presents classification results on 20Newsgroups dataset. The first three rows give the results as reported in earlier works. Raghu et al. [42], 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. Neural tensor skip-gram model (NTSG) [43] extends the idea of a skip-gram model for obtaining document embeddings, whereas sparse composite document vector (SCDV) [36] exploits pre-trained word embeddings to obtain sparse document embeddings. The authors in [36] have shown superior classification results as compared to paragraph vector, LDA, NTSG, and other systems. The next rows in Table III present our baselines and proposed systems. We see that the topic ID systems based on Bayesian SMM and logistic regression is better than many other models, except for the discriminatively trained CNN. We can also see that all the Bayesian SMM based systems are consistently better than variational auto encoder inspired NVDM.
TABLE II: Comparison of results on Fisher test sets, from earlier published works, our baselines and proposed systems.

| Systems     | Model  | Classifier | Manual transcriptions | Automatic transcriptions |
|-------------|--------|------------|-----------------------|--------------------------|
| Prior works | BoW    | NB         | 87.61                 | -                        |
|             | TF-IDF | LR         | 86.59                 | 0.93                     |
|             | ULMFiT | MLP        | 86.41                 | 0.50                     |
| Our Baseline|        |            |                       |                          |
|             | ℓ₁ SMM | LR         | 86.81                 | 0.91                     |
|             | ℓ₁ SMM | GLC        | 85.17                 | 1.64                     |
|             | NVDM   | LR         | 81.16                 | 0.94                     |
|             | NVDM   | GLC        | 84.47                 | 1.25                     |
|             | NVDM   | GLCU       | 83.96                 | 0.93                     |
| Proposed    |        |            |                       |                          |
|             | Bayesian SMM | LR  | 89.91                 | 0.89                     |
|             | Bayesian SMM | GLC | 89.47                 | 1.05                     |
|             | Bayesian SMM | GLCU| 89.54                 | 0.68                     |

TABLE III: Comparison of results on 20Newsgroups.

| Systems     | Model  | Classifier | Accuracy (%) | CE |
|-------------|--------|------------|--------------|----|
| Prior works | CNN    | -          | 86.12        | -  |
|             | SCDV   | SVM        | 84.60        | -  |
|             | NTSG-1 | SVM        | 82.60        | -  |
| Our Baselines|        |            |              |    |
|             | ℓ₁ SMM | LR         | 82.01        | 0.75 |
|             | ℓ₁ 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 from 20Newsgroups inferred using Bayesian SMM.

E. Uncertainty in document embeddings

The uncertainty captured in the posterior distribution of document embeddings correlates strongly with size of the document. The trace of the covariance matrix of the inferred posterior distributions gives us the notion of such a correlation. Fig. 7 shows an example of uncertainty captured in the embeddings. Here the Bayesian SMM was trained on 20Newsgroups with an embedding dimension of 100. The topic label uncertainty is not explicitly encoded in the posterior distributions; as it was an unsupervised model.

VIII. CONCLUSIONS AND FUTURE WORK

We have presented a generative model for learning document representations (embeddings) and their uncertainties. We showed that our model achieved superior perplexity results on the standard 20Newsgroups and Fisher datasets when compared to neural variational document model. Next, we have shown that the proposed model is robust to over-fitting and unlike in SMM and NVDM, it does not require any performance monitoring system 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. There exists other scoring mechanisms that exploit the uncertainty in embeddings [44]. Using simple linear classifiers on the obtained document embeddings, we achieved superior classification results on Fisher speech data and comparable results on 20Newsgroups text data. 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 in a straightforwardly for 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.
It is convenient to have the following derivatives:

\[ q(w) = N(w | \nu, \text{diag}(\exp(2\varsigma))) \]  
(42)

The lower bound for a single document is:

\[
\mathcal{L}_d \approx -\frac{1}{2} \left[ \lambda \text{tr}(\text{diag}(\exp(2\varsigma))) - \log(\text{diag}(\exp(2\varsigma))) \right] \\
- K \log \lambda + \lambda \nu^T \nu - K \\
+ \sum_{i=1}^{V} x_i \left[ (m_i + t_i \nu) \\
- \frac{1}{R} \sum_{r=1}^{R} \log \left( \sum_{j=1}^{V} \exp(m_j + t_j g(\epsilon_r)) \right) \right],
\]

where

\[ g(\epsilon) = \nu + \text{diag}(\exp(\varsigma)) \tilde{\epsilon}. \]  
(44)

It is convenient to have the following derivatives:

\[
\frac{\partial g(\epsilon)}{\partial \nu} = I.
\]  
(45)

\[
\frac{\partial (t_i g(\epsilon))}{\partial \varsigma} = \text{diag}(t_i^T \text{diag}(\exp(\varsigma))) \text{diag}(\tilde{\epsilon}) \\
= t_i^T \exp(\varsigma) \odot \tilde{\epsilon}.
\]  
(46)

Derivatives of the parameters of variational distribution:

Taking derivative of the objective function (Eq. (43)) with respect to \( \nu \) and using Eq. (45):

\[
\frac{\partial \mathcal{L}_d}{\partial \nu} = -\lambda \nu + \sum_{i=1}^{V} x_i \left[ t_i^T - \frac{1}{R} \sum_{r=1}^{R} t_i^T t_r \theta_{kr} \exp(m_k + t_k g(\epsilon_r)) \right] \\
+ \left[ \sum_{i=1}^{V} x_i t_i^T - \frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} \theta_{kr} x_k \right] - \lambda \nu
\]  
(48)

\[
\nabla \nu = \left[ \sum_{i=1}^{V} t_i^T (x_i - \frac{1}{R} \sum_{r=1}^{R} \theta_{kr} x_k) \right] - \lambda \nu.
\]  
(50)

Taking the derivative of objective function (Eq. (43)) with respect to \( \varsigma \) and using Eq. (46):

\[
\frac{\partial \mathcal{L}_d}{\partial \varsigma} = -\frac{1}{2} \left[ 2\lambda \exp(2\varsigma) - 2I \right] \\
+ \sum_{i=1}^{V} x_i \left[ -\frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} t_k^T \theta_{kr} \exp(m_k + t_k g(\epsilon_r)) \right] \\
+ \left[ \sum_{i=1}^{V} x_i \sum_{r=1}^{R} \sum_{k=1}^{V} \theta_{kr} t_k^T \exp(\varsigma) \odot \tilde{\epsilon} \right] - \omega \text{sign}(t_k)
\]  
(51)

\[
\nabla \varsigma = -\frac{1}{2} \left[ 2\lambda \exp(2\varsigma) - 2I \right] \\
- \left[ \sum_{i=1}^{V} x_i \sum_{r=1}^{R} \sum_{k=1}^{V} \theta_{kr} t_k^T \exp(\varsigma) \odot \tilde{\epsilon} \right] \\
+ \left[ \sum_{i=1}^{V} x_i \sum_{r=1}^{R} \sum_{k=1}^{V} \theta_{kr} (\sum_{j=1}^{V} \exp(m_j + t_j g(\epsilon_r))) \right] \\
- \omega \text{sign}(t_k)
\]  
(52)

Derivatives of the model parameters:

Taking the derivative of complete objective Eq. (47) with respect to a row \( t_k \) from matrix \( T \):

\[
\frac{\partial \mathcal{L}}{\partial t_k} = \sum_{d=1}^{D} \sum_{i=1}^{V} x_{di} \left[ (m_i + t_i \nu_d) \\
- \frac{1}{R} \sum_{r=1}^{R} \log \left( \sum_{j=1}^{V} \exp(m_j + t_j g(\epsilon_r)) \right) \right] \\
- \omega \sum_{i=1}^{V} t_i \|t_i\|_1
\]  
(53)

\[
\nabla t_k = \sum_{d=1}^{D} \left[ x_{dk} \nu_d^T - \sum_{i=1}^{V} x_{di} \frac{1}{R} \sum_{r=1}^{R} \theta_{dkr} g(\epsilon_r) \right] \\
- \omega \text{sign}(t_k)
\]  
(55)

Here, sign operates element-wise.
APPENDIX B
EM ALGORITHM FOR GLCU

E-STEP:

Obtaining the posterior distribution of latent variable
\( p(y_d | \nu_d, \Theta) \). Using the results from [27] (p. 41, Eq. (358)):

\[
\log p(y_d | \nu_d, \Theta) = \log p(\nu_d | y_d) + \log p(y_d) - \log p(\nu_d)
\]

\[
= \log \mathcal{N}(\nu_d | \mu_d + y_d, D^{-1})
\]

\[
+ \log \mathcal{N}(y_d | 0, \Gamma_d^{-1}) + \text{const}
\]

\[
= -\frac{1}{2} (\nu_d - (\mu_d + y_d))^T D (\nu_d - (\mu_d + y_d))
\]

\[
- \frac{1}{2} y_d^T \Gamma_d y_d + \text{const}
\]

\[
= -\frac{1}{2} (y_d - (\nu_d - \mu_d))^T D (y_d - (\nu_d - \mu_d))
\]

\[
- \frac{1}{2} y_d^T \Gamma_d y_d + \text{const}
\]

\[
= \mathcal{N}(y_d | u_d, V_d^{-1})
\]

where \( u_d \) is simplified as:

\[
u_d = (D + \Gamma_d)^{-1} (D (\nu_d - \mu_d) + \Gamma_d 0)
\]

\[
= [D^{-1} (D + \Gamma_d)]^{-1} (\nu_d - \mu_d)
\]

resulting in:

\[
u_d = [I + D^{-1} \Gamma_d]^{-1} (\nu_d - \mu_d)
\]

\[
V_d = D + \Gamma_d
\]

M-STEP:

Maximizing the auxiliary function

\[
\Theta^{\text{new}} = \arg \max_{\Theta} Q(\Theta, \Theta^{\text{old}})
\]

\[
q(y) = p(y | w, \Theta^{\text{old}}).
\]

Using the results from [27] (p. 43, Eq. (378)), the auxiliary function \( Q(\Theta, \Theta^{\text{old}}) \) is computed as:

\[
Q(\Theta, \Theta^{\text{old}}) = \mathbb{E}_q[\sum_{d=1}^{D} \log p(\nu_d, y_d)]
\]

\[
= \sum_{d=1}^{D} \mathbb{E}_q[\log p(\nu_d | y_d)] + \mathbb{E}_q[\log p(y_d)]
\]

\[
= \sum_{d=1}^{D} \mathbb{E}_q[\log \mathcal{N}(\nu_d | \mu_d + y_d, D^{-1})] + \text{const}
\]

\[
= \frac{D}{2} \log |D| - \frac{1}{2} \sum_{d=1}^{D} \left[ \mathbb{E}_q[(\nu_d - (\mu_d + y_d))^T D (\nu_d - (\mu_d + y_d))] + \text{const} \right]
\]

\[
= \frac{D}{2} \log |D| - \frac{1}{2} \sum_{d=1}^{D} \left[ \text{tr}(DV_d^{-1}) + (u_d - (\nu_d - \mu_d))^T D (u_d - (\nu_d - \mu_d)) \right]
\]

Maximizing the auxiliary function \( Q \) with respect to model parameters \( \Theta = \{M, D\} \)

Taking derivative with respect to each column \( \mu_{\ell} \) in \( M \) and equating it to zero:

\[
\frac{\partial Q}{\partial \mu_{\ell}} = -\frac{1}{2} \sum_{d \in I_{\ell}} [(u_d - (\nu_d - \mu_{\ell}))^T D (u_d - (\nu_d - \mu_{\ell}))]
\]

\[
= -\frac{1}{2} \sum_{d \in I_{\ell}} 2D (\mu_{\ell} - (\nu_d - u_d))
\]

\[
= -D \left( \sum_{n \in I_{\ell}} (\mu_n - \sum_{n' \neq n} (\nu_d - u_d)) \right)
\]

\[
\mu_{\ell} = \frac{1}{|I_{\ell}|} \sum_{n \in I_{\ell}} (\nu_d - u_d)
\]

Taking derivative with respect to shared precision matrix \( D \) and equating it to zero:

\[
\frac{\partial Q}{\partial D^{-1}} = \frac{D}{2} D^{-1} - \frac{1}{2} \left( \sum_{d=1}^{D} V_d^{-1} \right)^T
\]

\[
- \frac{1}{2} \left( \sum_{d=1}^{D} (u_d - (\nu_d - \mu_{\ell}))(u_d - (\nu_d - \mu_{\ell}))^T \right)
\]

\[
D^{-1} = \frac{1}{D} \left[ \sum_{d=1}^{D} V_d^{-1} + \sum_{d=1}^{D} (u_d - (\nu_d - \mu_{\ell}))(u_d - (\nu_d - \mu_{\ell}))^T \right]
\]

ACKNOWLEDGMENT

This work was supported by the U.S. DARPA LORELEI contract No. HR0011-15-C-0115. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work was also supported by Technology Agency of the Czech Republic project No. TJ01000208 "NOSIICI" and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602”

REFERENCES

[1] X. Wei and W. B. Croft, “LDA-based document models for ad-hoc retrieval,” in Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 2006, pp. 178–185. [Online]. Available: https://dl.acm.org/citation.cfm?id=1148204

[2] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” in IEEE Spoken Language Technology Workshop, December 2012, pp. 234–239. [Online]. Available: https://ieeexplore.ieee.org/document/6424228

[3] J. Wintride and S. Khudanpur, “Limited resource term detection for effective topic identification of speech,” in IEEE ICASSP, May 2014, pp. 7118–7122. [Online]. Available: https://ieeexplore.ieee.org/document/6854981

[4] X. Chen, T. Tan, X. Liu, P. Lanchantin, M. Wan, M. J. F. Gales, and P. C. Woodland, “Recurrent neural network language model adaptation for multi-genre broadcast speech recognition,” in Proc. Interspeech, ISCA, September 2015, pp. 3511–3515. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2015/papers/i15_3511.pdf
[44] N. Brümmer, A. Silnova, L. Burget, and T. Stafylakis, “Gaussian meta-embeddings for efficient scoring of a heavy-tailed PLDA model,” in Proc. Odyssey 2018 The Speaker and Language Recognition Workshop, 2018, pp. 349–356. [Online]. Available: [http://dx.doi.org/10.21437/Odyssey.2018-49](http://dx.doi.org/10.21437/Odyssey.2018-49)