Generative models for learning document representations along with their uncertainties

Thesis Submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Santosh Kesiraju
201150883
santosh.k@research.iiit.ac.in
Speech Processing Lab

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

INDIA
February 2021

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Generative models for learning document representations along with their uncertainties**" by Santosh Kesiraju, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____
Date

_____
Advisers: Dr. Suryakanth V Gangashetty
Dr. Lukáš Burget

Dedicated to my teachers

Acknowledgements

This thesis would not have been possible without the constant support from my advisers Dr. Suryakanth V Gangashetty and Dr. Lukáš Burget. I am amazed by the patience, calmness and consistency in Dr. Lukáš, with whom, the many and very long technical discussions were always productive. I express my sincere thanks to Dr. Suryakanth who was very supportive during challenging times.

I was lucky to be a part of two well known research labs in the area of speech and language technologies. The first one, Speech lab at IIIT-H lead by Prof. Bayya Yegnanarayana, one of the greatest teachers I have even seen. His lectures in the classroom and advices in the lab always ignite the mind. The second one is Speech@FIT, Brno University of Technology (BUT), lead by Prof. Jan "Honza" Černocký, one who wears many hats.

I express my gratitude to Dr. Kishore Prahallad for being my adviser during the initial years of my PhD, and helping me improve critical thinking and research methodologies. I would like to thank former Dean R&D Prof. Vasudeva Varma for supporting the collaboration between IIIT-H and BUT. I am deeply grateful to Tata Consultancy Services (TCS) for providing a 4-year scholarship for my PhD studies. I would like express my thanks to MediaEval organizers/chairs Martha Larson, Mohammad Soleymani and Xavier Anguera.

I would also like to thank my friends from IIIT-H, with whom I've spent many memorable moments and had several intellectual discussions: Sreedhar, Bhargav, Nivedita, Siva, Baji, Vishala, Anand, Sudarsana, Mohan, Naresh, Buchi Babu, Chaitanya, and Gautam. I would like to thank my friends and colleagues from BUT, where we constantly collaborate and learn from each other: Karthick, Olda, Ondra N, Martin K, Mirko, Ondra G, Lucas, Mireia, Pavel, Vlada, Fede, Katka, Kate, Karel V, Karel B, Hari, Franta, Martin F, and Petr. A special thanks Ms. Renata Kohlová who assited me with visa, residency, stay and other logistics in the Czechia. A special thanks to Igor Szöke who actually gave me an opportunity to intern at Speech@FIT, BUT.

I would like to express my sincere gratitude to Dr. Najim Dehak and Dr. Sanjeev Khudanpur for the internship and the opportunity to participate in the JSALT 2016 Workshop at Johns Hopkins University (JHU). I would like to thank my friends Raghu (JHU) and Harish (who also spent an year at BUT) for their collaborations.

I would like to deeply thank my parents for their constant support and understanding. A

special thanks to my brother who actually encouraged and inspired me to pursue a research career. I would like to thank my friend, my wife, Michaela for believing and being there for me.

I would also like to thank my cousin Vijay, and nephew Kautilya for their help during the final days of my dissertation.

Abstract

Majority of speech and natural language processing applications rely on word and document representations (or embeddings). The document embeddings encode semantic information which makes them suitable for tasks such as topic identification (document classification), topic discovery (document clustering), language model adaptation, and query-based document retrieval. These embeddings are usually learned from widely available un-labelled data; hence generative or probabilistic topic models which aim to capture the distribution of data are suitable.

Although there exist several probabilistic and neural network-based topic models to learn these embeddings, they often ignore to capture the uncertainty in the estimated embeddings. Thus, any error in the estimation of these embeddings affects the performance in downstream tasks. The uncertainty in the embeddings is usually due to shorter, ambiguous or noisy sentences/documents.

This thesis presents model(s) for learning to represent document embeddings in the form of Gaussian distributions, thereby encoding the uncertainty in their covariances. Further, these learned uncertainties in embeddings are exploited by the proposed generative Gaussian linear classifier for topic identification.

This thesis proposes to use subspace multinomial model (SMM), a simple log-linear model for learning document embeddings. Experiments on *20Newsgroups* text corpus show that the embeddings extracted from SMM are superior when compared to popular topic models such as latent Dirichlet allocation, sparse topical coding in topic identification and document clustering tasks. Using the variational Bayes framework on SMM, the model is able to infer the uncertainty in document embeddings, represented by (posterior) Gaussian distributions. Additionally, the common problem of intractability which appears while performing variational inference in mixed-logit models is addressed using Monte Carlo sampling via the re-parametrization trick. The resulting Bayesian SMM achieves state-of-the-art perplexity results on *20Newsgroups* text and *Fisher* speech corpora. The proposed generative classifier exploits the learned uncertainty in the document embeddings; and achieves state-of-the-art classification results on the aforementioned corpora as compared to other unsupervised topic and document models.

Furthermore, this thesis presents a multilingual extension of the Bayesian SMM for zero-shot cross-lingual topic identification. The proposed model achieves superior classification results

when compared to the systems based on multilingual word embeddings and neural machine translation inspired sequence-to-sequence bidirectional long-short term memory models.

Keywords: Variational inference, unsupervised methods, generative models, document modelling, topic modelling, embeddings, i-vectors, uncertainties, topic identification, multilingualism, zero-shot learning.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

a, α	Lower case symbols denote scalars
$\boldsymbol{b}$, $\boldsymbol{\beta}$	Lower case bold faced symbols denote column vectors unless otherwise specified
$\boldsymbol{C}$, $\boldsymbol{\Gamma}$	Upper case bold faced symbols denote matrices
$\boldsymbol{C}^{-\mathsf{T}}$	$(\boldsymbol{C}^{-1})^{\mathsf{T}}$ Inverse and Transpose of a matrix
$\text{diag}(\boldsymbol{b})$	Diagonal square-matrix with elements from the vector $\boldsymbol{b}$ along the diagonal
Dir	Dirichlet distribution
Gamma	Gamma distribution
Multi	Multinomial distribution
$\mathcal{N}$	Gaussian distribution
$\Gamma(.)$	Gamma function
$\nabla_{\boldsymbol{w}}\mathcal{L}$	Gradient of $\mathcal{L}$ with respect to $\boldsymbol{w}$
$\tilde{\nabla}_{\boldsymbol{t}}\mathcal{L}$	Sub-gradient of $\mathcal{L}$ with respect to $\boldsymbol{t}$
$\mathcal{P}_{\mathcal{S}}$	Sign projection
$\mathcal{P}_{\mathcal{O}}$	Orthant projection
$\triangle^{\mathsf{n}}$	n-Simplex
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integers
$\mathbb{Z}^*$	Set of non-negative integers
$\|\|\,.\,\|\|_p$	p-norm

Chapter 1

Introduction

This thesis presents new methods for modelling text and spoken documents. It involves obtaining low-dimensional (compact) representations (or embeddings) of documents. These representations elicit the latent semantic relations present among co-occurring words in a sentence or "bag-of-words" from a document. Learning these representations have a wide range of applications in information retrieval, speech and language processing applications such as topic identification/discovery, language model adaptation, sentiment analysis, query-based document retrieval and many more. Majority of the techniques for learning these representations are based on two complementary ideologies: (i) topic modelling, and (ii) word prediction. The former methods are primarily based on bag-of-words assumption and tend to capture higher-level semantics such as topics. The latter techniques capture lower-level semantics by exploiting the contextual information from words in a sequence (Mikolov et al., 2013; Pennington et al., 2014; Le and Mikolov, 2014). The nature of data and end application plays a role in the choice of these approaches.

On the other hand, there is a growing interest towards developing pre-trained language models, that are then fine-tuned for specific tasks such as document classification, question answering, named entity recognition, etc (Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019). Although these models achieve state-of-the-art results in several NLP tasks; they require enormous computational resources to train (Devlin et al., 2019).

The models and methods presented in this thesis mostly rely on the "bag-of-words" representation of document, and thus are more suitable for capturing higher-level semantics such as topics. These models are also seen as *generative models* for documents or *unsupervised topic models* that can be trained on largely available unlabelled data[1]. With the help of simple *linear classifiers*, the learned representations from these models are used for topic identification (ID). Note that these topic models are not same as the fancied neural network-based discriminative text classifiers (Zhang et al., 2015; Yang et al., 2016b). The latter are also able to learn internal representations of documents, but are constrained by relatively lower amounts of labelled

[1]Data without topic label annotations.

data. Moreover, adapting large-scale discriminative classifiers to newer data and classes requires re-training on the entire data, which might be computationally expensive.

Although topic models have existed for many years (Deerwester et al., 1990), the research in this domain is continuously evolving (Miao et al., 2016; Srivastava and Sutton, 2017). Of these, probabilistic topic models (PTM) are popular and tend to be preferred because of their interpretability (Blei, 2012) and structure which enables them to be integrated into other probabilistic models (Wallach, 2006). 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 modelled as a distribution over words in the vocabulary. For example, classical models such as latent Dirichlet allocation (LDA) (Blei et al., 2003) learns to represent documents and words in the form of discrete probability distributions, where as the models proposed in this thesis represent them in the form of Gaussian distributions. The advantages of the latter over former are discussed in Chapters 3, 4 and 5.

Recent works showed that auto-encoders can also be seen as generative models for images, text and speech (Kingma and Welling, 2014; Chung et al., 2015; Miao et al., 2016). Generative models allows us to incorporate prior information about the latent variables, and with the help of (stochastic) variational Bayes (VB) techniques (Bishop, 2006; Hoffman et al., 2013; Rezende et al., 2014), 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.

This thesis work builds on top of the works of Kockmann (2011) and Soufifar (2014), which were primarily based on subspace multinomial model (SMM) and its variants for various speech processing applications. SMM was originally proposed for modelling discrete prosodic features for the task of speaker verification (Kockmann et al., 2010) and latter, SMM and its extension subspace n-gram model (SnGM) were used for phonotactic language recognition (Soufifar et al., 2011, 2013).

Firstly, this thesis proposes to use SMM for learning document representations. By using ℓ_1 regularization over the model parameters in SMM and employing orthant-wise learning, we introduce sparsity into the model; which is one of the desired properties in text modelling. The document classification and clustering experiments on *20Newsgroups* corpus show that the document representations obtained from the proposed ℓ_1 SMM are superior to ones obtained from classical topic models such as latent Dirichlet allocation (Blei et al., 2003), non-negative matrix factorization (NMF) (Xu et al., 2003), and sparse topical coding (Zhu and Xing, 2011). However, the experimental analysis showed that the document embeddings extracted using SMM are prone to over-fitting, especially when the target documents are relatively short.

The shortcomings of SMM are addressed by employing a Bayesian framework and modelling the uncertainties. The proposed Bayesian SMM can learn to represent the documents in the

form of (posterior) Gaussian distributions, thereby encoding the uncertainty about the estimates in its covariance. This uncertainty gives a notion of how well the embeddings represent the original document. Moreover, this uncertainty is exploited during training the classifiers for downstream tasks such as topic identification.

The experiments on *Fisher* speech and *20Newsgroups* text corpora show that the proposed Bayesian SMM fits the unseen test data better and achieves state-of-the-art perplexity results (Kesiraju et al., 2020a). Further, a generative Gaussian linear classifier is proposed, which can exploit the learned uncertainty in the document embeddings. The classification experiments on both the aforementioned datasets show that the proposed model together with the classifier is robust to over-fitting and achieves superior classification results when compared to other unsupervised topic models (Miao et al., 2016), and comparable results to the state-of-the-art discriminative models (Howard and Ruder, 2018; Pappagari et al., 2018).

Next, the Bayesian SMM is extended to the multilingual scenario, which aims to learn language-agnostic document embeddings (Kesiraju et al., 2020b), that are helpful in zero-shot cross-lingual topic identification. The experiments on *Europarl* (Koehn, 2005) and *Reuters multilingual news* (MLDoc) corpora show that the proposed model is superior to multilingual word embedding based systems and sequence-to-sequence bi-directional long short-term memory (BiLSTM) network based systems (Schwenk and Douze, 2017) in majority of the transfer directions.

The proposed extensions and variants of SMM are used for learning document embeddings that are used in downstream tasks such as (cross-lingual) topic identification (Kesiraju et al., 2020a), and language model adaptation (Beneš et al., 2018). However, their performance in phonotactic language recognition and in other applications do not come under the scope of this thesis. Towards the end, this thesis presents models that can (i) exploit both the labelled and unlabelled data; thus making the best use of generative and discriminative the approaches, (ii) learn document embeddings by exploiting contextual information from words with-in a sentence, thus capturing lower-level semantics.

1.1 Applications of document representations

This section briefly outlines some of the applications and tasks relying on document embeddings. The experiments in this thesis focus on topic identification and document clustering.

1. **Topic identification** (ID) requires to classify a given set of documents into one of the pre-selected topics or categories. The analogous task in the unsupervised scenario is document clustering; which requires to cluster the documents so that each cluster (ideally) represents a single topic. This clustering can also be seen as **topic discovery**; where a large corpus of

documents (e.g. scientific articles from JSTOR[2] or e-books from Project Gutenberg[3]) can be analysed (Blei and Lafferty, 2005) based on the discovered topics; where each topic is represented by a mixture of words from the vocabulary. The key element in approaching either of the tasks is by learning a (low dimensional) semantic-rich representation for every document. Having such a compact representation further allows us to train and adapt simple (linear) classifiers for topic ID. For example, if the document embeddings are Gaussian distributed, one can use simple Gaussian linear classifier (GLC) for topic ID. Moreover, GLC can be easily adapted to newer data and classes (topics) without requiring to re-train on the entire data.

2. **Zero-shot cross-lingual topic ID** requires to training a classifier in source (SRC) language which is then used to classify documents (samples) from target (TAR) language. Model selection and hyper-parameter tuning is done based on the evidence from source language only. This problem is approached by learning a common embedding space for multiple (say, L number of) languages (Ammar et al., 2016; Schwenk and Li, 2018; Ruder et al., 2019). This common embedding space is learnt by exploiting parallel dictionary or parallel sentences among the L languages. Such a parallel data is not required to have topic labels. A classifier is then trained on the embeddings from a source (SRC) language (one from the L languages) that has topic labels. The same classifier is then and used to classify the embeddings extracted for test data, which can be from any of the L target (TAR) languages. The underlying assumption here is that the embeddings carry semantic concept(s), independent of language, enabling cross-lingual transferability (SRC $\rightarrow$ TAR). Hence, the reliability of this *scheme* solely depends on quality of the embedding space. Note that the amount of available data for training the classifier could be limited and different from the parallel data, which is also the case for the experiments presented in this thesis (Chapter 7).

3. **Language model (LM) adaptation**: LM is one the major components in automatic speech recognition (ASR), machine translation, parts-of-speech tagging, hand writing and optical character recognition systems. For example, in the case of ASR, adapting a language model to a specific domain or context helps disambiguating *homophones*[4] and similar sounding phrases.

   There is a wide range of context specific information such as topics, geographic location (Chelba et al., 2015), personal profile (in case of ASR on smart phones and devices) that are used for adapting language models. Topic information can be easily incorporated into LMs. For example, Wallach (2006) used a hierarchical Bayesian model that incor-

[2] https://www.jstor.org/
[3] https://www.gutenberg.org/
[4] Words that sound same but spell different.

porates n-gram statistics and latent topic variables. This structured integration requires the use of probabilistic models and provides an elegant interpretation.

Alternatively, one can use document of word embeddings obtained from a topic model as additional feature vectors while training a language model. Mikolov and Zweig (2012); Chen et al. (2015) have used latent Dirichlet allocation (Blei et al., 2003) to fit a topic model and then used the inferred word representations as additional feature vectors while training a recurrent neural network based LM. Jin et al. (2015) clustered document embeddings extracted from paragraph vector (Le and Mikolov, 2014), and trained a cluster (topic) specific LM. This cluster-specific LM was then interpolated with a global LM that was trained on entire data. This requires multi-pass decoding of speech signal to figure out which topic-specific LM should be used for interpolation. The models discussed in this thesis were used in an effective way[5] for adapting feed-forward neural network-based language models. More details are given in (Beneš et al., 2018).

4. **Query based document retrieval** is another application, where similarity among words and documents play a pivotal role (Wintrode and Khudanpur, 2014). When using discrete probabilistic representations of documents and words (such as from LDA), one can compute symmetric Kullback Leibler divergence between words and document embeddings to find relevant documents (Wei and Croft, 2006). For other kinds of non probabilistic representations, euclidean or cosine distance is preferred.

1.2 Challenges

In the growing era of smart phones, IoT devices and other gadgets in the human computer interaction (HCI) scope (for example, Google's voice assistant, Apple Siri, Amazon Alexa, Microsoft Cortana), there is a large amount of speech and text (multi-modal and multilingual) data being gathered and processed. New kinds of data and newer ways of HCI, creates new challenges and hence a need for newer models and algorithms. The following section briefly outlines the nature of data in terms of content, labelling, and assumptions (or simplifications); thereby highlighting some of the major challenges.

1.2.1 Spoken vs. text documents

Spoken audio comes from a variety of sources covering a wide range of content involving conversation telephone speech (data from call centres), recordings of meetings, broadcast news, video lectures, talks, audio-video from social media, and, pod casts. In this thesis, spoken text is referred as the text that is transcribed from spoken audio either using ASR system or by humans (referred as manual transcription). The word statistics in the spoken text vary, depending

[5] Work done in collaboration (Beneš et al., 2018).

on the nature of the spoken audio. For example, a conversational speech or speech from a meeting recording contains many dis-fluencies and irregularities as compared to broadcast news or recorded lectures. The latter are more structured and is mostly read-speech.

There are other challenges involved in processing spoken audio that are caused by speaker, channel and environmental variabilities. These are usually addressed by the ASR community and is not in the scope of this thesis.

Text documents origin from various sources such as news, blogs, tweets, product reviews or social news aggregation platforms. For example, text from news or blogs is mostly structured and grammatically correct, where as text from "tweets" or "product reviews" is more likely to contain un-conventional word usages, internet slangs, emoticons and other symbols.

The desired property of topic models is to achieve robust (word and) document representations irrespective of the nature of content and the kind of tokenisation.

1.2.2 Unlabelled and labelled data

It is evident that the amount of data being created and stored is increasing at an astronomical rate[6] (including the data on Wikipedia[7]). The amount of labelled (topic, sentiment, category labels, etc.) data, however remains relatively very small, as it requires human effort that is time consuming and expensive. These labelled data could be used to train discriminative classifier models which could be used to categorize the unlabelled data. However, adapting large-scale discriminative classifiers to newer classes (or topics) requires re-training which could be computationally expensive. Alternatively, semi-supervised approaches or hybrid models (Lasserre, 2008) can use both the labelled and unlabelled data. The proposed models is this thesis could be easily translated into hybrid models (Chapter 8) thus exploiting both the labelled and unlabelled data.

1.2.3 Bag-of-words

Bag-of-words is a simplified view of a document, where the word order is ignored and a document is represented by a vector of word occurrences; thereby significantly reducing the size of the document. Majority of the topic models are built on top of this bag-of-words simplification, and are capable of capturing higher level semantics such as topics. However, topic models trained with such simplification on a very short documents (few sentences) can lead to inefficient estimates of document and word representations (embeddings). In such scenarios, Bayesian topic models are useful as the uncertainty in the document representations is captured by the posterior distribution. Bag-of-words is not an optimal choice for tasks like sentiment analysis, where the word order plays a significant role. The semantic meaning of the word in

[6] https://www.internetlivestats.com/

[7] https://en.wikipedia.org/wiki/Wikipedia:Statistics

its shorter context is lost in "bag-of-words" model. In such scenarios, approaches based on language modelling or word prediction can be very useful.

The models and methods presented in this thesis mostly rely on the "bag-of-words" simplification of document, and thus are more suitable for capturing higher-level semantics such as topics. It is possible to extend these models for obtaining sentence representations by exploiting information from the n-gram contexts (Chapter 8).

1.3 Overview and original contributions

An overview of rest of the thesis along with the original contributions are as follows:

1. Chapter 4 presents subspace multinomial model (SMM) for learning document representations. A variant of SMM based on ℓ_1 regularization of its parameters was proposed, which is superior to other unsupervised models such as latent Dirichlet allocation and sparse topical coding, on topic identification (ID) and document clustering tasks. These details are presented in . The analyses of learned document representations, and its nature to over-fitting motivated the need for Bayesian modelling.

2. Chapter 5 presents Bayesian subspace multinomial model (Bayesian SMM), which aims to capture the uncertainty of document representations. The Bayesian framework introduces an additional problem of intractability, which commonly appears while performing Bayesian inference in mixed-logit models. This problem of intractability is resolved with the Monte Carlo approximation via re-parametrization trick. Towards the end, the experiments show that the proposed Bayesian SMM can indeed learn document representations along with their uncertainties. The model achieves state-of-the-art perplexity results on *20Newsgroups* text and *Fisher* speech corpora under limited and full vocabulary settings, when compared to other document models.

3. Motivated by the encoded information about the uncertainty of document representations (posterior distributions), a generative classifier is proposed that can exploit the uncertainty. The proposed classifier is called Gaussian linear classifier with uncertainty (GLCU) and is presented in Chapter 6. The experiments show that the proposed systems are robust to over-fitting on unseen text data and achieves state-of-the-art classification accuracy on topic identification tasks on *Fisher* speech and *20Newsgroups* text corpora.

4. Chapter 7 presents an extension of Bayesian SMM which can learn language-agnostic document embeddings with the help of L-way parallel data. The experiments on *Europarl* and *Reuters multilingual news* corpora show that the proposed multilingual Bayesian model is superior to multilingual word embedding based systems and sequence-to-sequence bi-directional long short-term memory based systems. Moreover, our system which is

trained with less than a million sentences on a single NVIDIA Tesla P-100 GPU under 24 hours, performs competitively against state-of-art BiLSTM system trained on 223 million sentences, that takes about 5 days on 16 NVIDIA V100 GPUs (Artetxe and Schwenk, 2019).

5. Chapter 8 presents two variants of Bayesian SMM that can (i) use both labelled and un-labelled data, and (ii) learn document/sentence embeddings by exploiting contextual n-grams from sentences.

6. The conclusions of the thesis and directions for future research are discussed in Chapter 9.

7. The list of accepted publications and current articles in review is given in § Publications.

8. The source code for the proposed models is available for public[8].

[8] https://github.com/skesiraju

Chapter 2

Evaluation methods

This chapter outlines the methods or tasks that are used in this thesis to evaluate document representations and topic models.

1. **Topic identification** is a classification task, where the evaluation is based on classification accuracy and cross-entropy loss on the test set. 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.

   Consider a dataset with N number of samples and M number of classes. Let $\boldsymbol{y}_n$ be the one-hot encoding vector representing the true class k, for a sample n, i.e.

   $$y_{ni} = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise } (i = 1 \dots M, i \neq k). \end{cases} \tag{2.1}$$

   Let $\hat{\boldsymbol{y}}_n$ be the vector of predicted posterior probabilities of M class labels for a sample n, such that $\sum_{i=1}^{M} \hat{y}_{ni} = 1$. Then, the cross-entropy between true classes $\mathcal{Y}$ and predicted class probabilities $\hat{\mathcal{Y}}$ for the entire dataset is

   $$H(\mathcal{Y}, \hat{\mathcal{Y}}) = -\sum_{n=1}^{N} \sum_{i=1}^{M} y_{ni} \ln \hat{y}_{ni} \tag{2.2}$$

2. **Document clustering** is an unsupervised task which is commonly evaluated by computing normalized mutual information (NMI) (Manning et al., 2008) between the true classes $\mathcal{Y}$ and the obtained clusters $\mathcal{C}$:

   $$\text{NMI}(\mathcal{Y}, \mathcal{C}) = \frac{2 \times I(\mathcal{Y}; \mathcal{C})}{H[\mathcal{Y}] + H[\mathcal{C}]}, \tag{2.3}$$

   where $H[\mathcal{Y}]$ represents entropy of class labels which can be calculated prior to the clustering. $H[\mathcal{C}]$ represents entropy of clusters, and $I(\mathcal{Y}; \mathcal{C})$ is the mutual information between $\mathcal{Y}$ and $\mathcal{C}$.

Consider a dataset of N samples comprising of M number of (true) class labels. Let K denote the number of clusters obtained from a clustering algorithm. Then, entropy of class labels:

$$H[\mathcal{Y}] = -\sum_{m=1}^{M} p(\mathcal{Y}_i) \ln p(\mathcal{Y}_i), \tag{2.4}$$

where $p(\mathcal{Y}_i)$ represents prior of class i, which is estimated using maximum likelihood approach, i.e.

$$p(\mathcal{Y}_i) = \frac{|\mathcal{Y}_i|}{N}, \tag{2.5}$$

where $|\mathcal{Y}_i|$ is the number of samples in class i.

Similarly, entropy of cluster labels

$$H[\mathcal{C}] = -\sum_{i=k}^{K} p(\mathcal{C}_k) \ln p(\mathcal{C}_k). \tag{2.6}$$

The mutual information

$$I(\mathcal{Y}; \mathcal{C}) = \sum_{k=1}^{K} \sum_{i=1}^{M} p(\mathcal{C}_k \cap \mathcal{Y}_i) \ln \frac{p(\mathcal{C}_k \cap \mathcal{Y}_i)}{p(\mathcal{C}_k) p(\mathcal{Y}_i)}, \tag{2.7}$$

where $p(\mathcal{C}_k \cap \mathcal{Y}_i)$ is estimated using maximum likelihood

$$p(\mathcal{C}_k \cap \mathcal{Y}_i) = \frac{|\mathcal{C}_k \cap \mathcal{Y}_i|}{N}, \tag{2.8}$$

$|\mathcal{C}_k \cap \mathcal{Y}_i|$ represents number of samples in the intersection of cluster k and class i.

3. **Perplexity** is inversely proportional to the log-probability 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 (Bengio et al., 2003; Jurafsky and Martin, 2009). Since topic models built on bag-of-words are equivalent to uni-gram language models, perplexity is seen as an intrinsic measure for topic models (Blei et al., 2003; Srivastava et al., 2013a; Miao et al., 2016).

   To evaluate probabilistic topic models, perplexity is computed on an unseen test data. It is computed in two ways:

   (a) As an average for a document $\boldsymbol{x}_d$ in a corpus of D documents according to:

   $$\text{PPL}_{\text{DOC}} = \exp\Big\{ -\frac{1}{D} \sum_{d=1}^{D} \frac{\ln p(\boldsymbol{x}_d)}{N_d} \Big\}, \tag{2.9}$$

   (b) Across the entire corpus of D documents as:

   $$\text{PPL}_{\text{CORPUS}} = \exp\Big\{ -\frac{\sum_{d=1}^{D} \ln p(\boldsymbol{x}_d)}{\sum_{d=1}^{D} N_d} \Big\} \tag{2.10}$$

   where N_d is the number of words in document d.

Chapter 3

Generative models for documents

This chapter presents an overview of generative models, Bayesian and approximate inference techniques. The variational Bayes (inference) framework is derived and explained, as it forms the basis for majority of the existing and proposed models discussed in this thesis.

Next, some of the popular probabilistic topic models (generative models for "bag-of-words" representation of documents) are discussed. These include latent Dirichlet allocation (LDA), correlated topic model (CTM), paragraph vector (PV-DBOW), neural variational document model (NVDM). The modelling assumptions, inference techniques along with their limitations are also discussed. This thesis proposes novel probabilistic topic models that aim to overcome these limitations.

3.1 Introduction to generative models

Probabilities play a central role in pattern recognition and machine learning. Generative models are probabilistic models that aim to capture (model) the distribution of data (and labels). Two broad kinds of generative models are discussed here: The first one aims to model distribution of data $p(\boldsymbol{x})$, the second is generative classifier that aims to model the joint distribution of data and corresponding class labels $p(\boldsymbol{x}, \boldsymbol{y})$. The discussion in this chapter primarily focuses on unsupervised models which aim to model $p(\boldsymbol{x})$, and the theoretical concepts derived can be extended in a straightforward way to the generative classifiers. Modelling the distribution of data has several advantages:

1. Reducing the dimensionality of data.

2. Generating synthetic data.

3. Interpolating missing data points.

4. Handling mismatched data conditions between training and test sets.

It is often convenient to represent probabilistic models in a graphical representation. A simple generative model is graphically depicted in Fig. 3.1. "The graphical model captures the *causal* process by which the observed data was generated (Bishop, 2006)". The graphical model does not provide any information regarding the probability distributions/densities of observed or latent variables. The following steps explain the generative process of graphical model from Fig. 3.1:

For every data point $n = 1 \dots N$:

A latent variable $\boldsymbol{z}_n$ is sampled from a probability distribution parametrized by $\boldsymbol{\alpha}$:

$$\boldsymbol{z}_n \sim p(\boldsymbol{z}_n \mid \boldsymbol{\alpha}). \tag{3.1}$$

$p(\boldsymbol{z}_n \mid \boldsymbol{\alpha})$ can be seen as prior distribution over latent variables.

Given the model parameters $\boldsymbol{\theta}$, every data point $\boldsymbol{x}_n$ is generated from a conditional distribution:

$$\boldsymbol{x}_n \sim p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}). \tag{3.2}$$

$p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta})$ is also called likelihood of the data.

The above generative process fully describes the probabilistic model depicted in Fig. 3.1, where, every data point $\boldsymbol{x}_n$ conditioned on latent variable $\boldsymbol{z}_n$ is assumed to be generated independent of other data points according to a stochastic process.

Two examples for the latent variable $\boldsymbol{z}$ are presented below:

1. $\boldsymbol{z}_n$ can be a K dimensional discrete variable in one-hot encoding[1] format, that represents a topic.

2. $\boldsymbol{z}_n$ can be a K dimensional continuous variable which encodes the topic information in terms of semantic correlations among the words.

In both cases, the generated $\boldsymbol{x}_n \in \mathbb{R}^V$ $(V \gg K)$ will be a topic-specific document. In the first case $\boldsymbol{z}_n$ can be seen as topic-label, whereas in the second case $\boldsymbol{z}_n$ can be seen an embedding for the document $\boldsymbol{x}_n$.

Given training data $\boldsymbol{X}$ comprising of N examples, where every row $\boldsymbol{x}_n$ $n = 1 \dots N$ corresponds to a single example; the model can be trained to obtain (estimate) the parameters $\boldsymbol{\theta}$. For any given unseen (test) data point, the posterior distribution over the latent variables $p(\boldsymbol{z}_n \mid \boldsymbol{x}_n)$ can be inferred. The obtained latent representations can be useful in classification and clustering tasks.

[1] A vector where only one element is 1, and the rest are zeros.

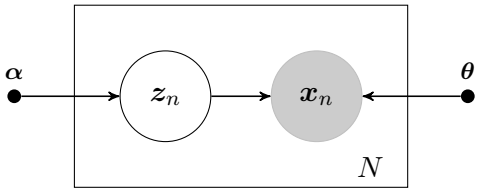


Figure 3.1: Graphical representation of a simple generative model. The rectangular plate with N represents the number of data points generated. $\boldsymbol{x}_n$ is a observed data sample (hence, shaded), where as $\boldsymbol{z}_n$ is a hidden (latent) variable. $\boldsymbol{\theta}$ represents model parameters, and $\boldsymbol{\alpha}$ is a hyper-parameter.

We begin with joint distribution of data $\boldsymbol{X}$ and latent variables $\boldsymbol{Z}$:

$$p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}, \boldsymbol{\alpha}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n \mid \boldsymbol{\theta}, \boldsymbol{\alpha}) \tag{3.3}$$

$$= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta})\, p(\boldsymbol{z}_n \mid \boldsymbol{\alpha}). \tag{3.4}$$

The joint distribution factorizes[2], and applying Bayes' rule to (3.4), the posterior distribution of a latent variable $\boldsymbol{z}$ can be written as[3]:

$$p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z} \mid \boldsymbol{\alpha})}{p(\boldsymbol{x} \mid \boldsymbol{\theta})} \tag{3.5}$$

$$= \frac{p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z} \mid \boldsymbol{\alpha})}{\int p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z} \mid \boldsymbol{\alpha}) \mathrm{d}\boldsymbol{z}}. \tag{3.6}$$

To obtain the posterior distribution, the integral in denominator of (3.6) needs to be computed. Depending on the functional forms of the probability distributions (assumptions of the generative model), this denominator can be intractable, i.e., if the likelihood function and prior are not conjugate to each other (Bishop, 2006). Notable examples with such intractability include Bayesian Gaussian mixture model, Bayesian logistic regression, latent Dirichlet allocation. In such cases where the true posterior is intractable, one can resort to variational inference.

3.1.1 Variational inference

The idea of variational inference (VI) or *variational Bayes* (VB) is to find a parametric probability distribution $q(\boldsymbol{z})$ that approximates the true posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$ by minimizing the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(q \,||\, p)$ from the approximate to the true posterior. Computing this KL divergence still requires a functional form of the true posterior distribution.

[2]Since we assumed every data point is i.i.d

[3]Omitting the suffix n for brevity.

There exists an alternative approach that avoids the computation of the true posterior. We proceed as follows:

First, the KL divergence term is expanded:

$$D_{\mathrm{KL}}(q \,||\, p) = -\int q(\boldsymbol{z}) \ln\left(\frac{p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha})}{q(\boldsymbol{z})}\right) \mathrm{d}\boldsymbol{z} \tag{3.7}$$

$$= -\int q(\boldsymbol{z}) \ln p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{z} - \underbrace{\left(-\int q(\boldsymbol{z}) \ln q(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}\right)}_{\mathrm{H}[q]} \tag{3.8}$$

$$= -\int q(\boldsymbol{z}) \ln p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{z} - \mathrm{H}[q], \tag{3.9}$$

where $H[q]$ is the differential entropy of $q(\boldsymbol{z})$.

Next, the log marginal is expressed as:

$$\ln p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \ln p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}, \boldsymbol{\alpha}) - \ln p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}) \tag{3.10}$$

$$= \ln p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}, \boldsymbol{\alpha}) \underbrace{\int q(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}}_{1} - \ln p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}) \underbrace{\int q(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}}_{1} \tag{3.11}$$

adding and subtracting $H[q]$ term; re-arranging, we get:

$$\ln p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \int q(\boldsymbol{z}) \ln p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}, \boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{z} + \mathrm{H}[q] \underbrace{- \int q(\boldsymbol{z}) \ln p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{z} - \mathrm{H}[q]}_{D_{\mathrm{KL}}(q \,||\, p)}. \tag{3.12}$$

Now, making use of (3.9),

$$\ln p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \underbrace{\mathbb{E}_q[\ln p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})] + \mathrm{H}[q]}_{\mathcal{L}(q)} + D_{\mathrm{KL}}(q \,||\, p). \tag{3.13}$$

Finally, log marginal is expressed as:

$$\boxed{\ln p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathcal{L}(q) \; + \; D_{\mathrm{KL}}(q \,||\, p)} \tag{3.14}$$

(3.14) is the standard variational Bayes formulation (Bishop, 2006), where log marginal of the data is expressed as the sum of $\mathcal{L}(q)$ and the KL divergence term $D_{\mathrm{KL}}(q \,||\, p)$. Given the observed data $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$, log marginal is a constant and $D_{\mathrm{KL}}(q \,||\, p)$ can be minimized by maximizing $\mathcal{L}(q)$. The KL divergence is always non-negative and is equal to zero only when $q(\boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\alpha})$. As noted earlier, the advantage of this formulation is that the KL divergence term need not be evaluated. The term $\mathcal{L}(q)$ is a functional[4] of $q(\boldsymbol{z})$ and acts

[4] Function of a function.

Algorithm 1: Variational Bayes EM algorithm

initialize prior belief $p(\boldsymbol{z} \mid \boldsymbol{\alpha})$
initialize model parameters $\boldsymbol{\theta}$
initialize variational distribution $q(\boldsymbol{z} \mid \boldsymbol{\beta})$
repeat
    `// VB E-step`
    compute gradients of $\mathcal{L}(q)$ w.r.t $\boldsymbol{\beta}$
    update variational distribution $q(\boldsymbol{z} \mid \boldsymbol{\beta})$
    `// VB M-step`
    compute gradients of $\mathcal{L}(q)$ w.r.t $\boldsymbol{\theta}$
    update model parameters $\boldsymbol{\theta}$
until *convergence or max_iterations*

a lower bound on log marginal of the data. Hence it is referred to as **e**vidence **l**ower **bo**und (ELBO) or variational lower bound and $q(\boldsymbol{z})$ is referred to as variational distribution.

Given this formulation, the goal of finding an approximate posterior $q(\boldsymbol{z})$ and the model parameters $\boldsymbol{\theta}$ is now converted into an optimization problem, where we need find such a $q(\boldsymbol{z} \mid \boldsymbol{\beta})$ (where $\boldsymbol{\beta}$ are the parameters of $q(\boldsymbol{z})$) and $\boldsymbol{\theta}$ that maximizes ELBO. This is achieved by the following VB expectation-maximization (EM) algorithm. An example is given in Algorithm 1.

In the `VB E-step`, the variational distribution $q(\boldsymbol{z} \mid \boldsymbol{\beta})$ is updated by keeping the model parameters $\boldsymbol{\theta}$ fixed. This moves $q(\boldsymbol{z} \mid \boldsymbol{\beta})$ closer to the true posterior, thus reducing the KL divergence $D_{\mathrm{KL}}(q \,||\, p)$. In the successive `VB M-step`, the model parameters ($\boldsymbol{\theta}$) are updated to better explain the observed data $\boldsymbol{x}$; this makes the $D_{\mathrm{KL}}(q \,||\, p)$ larger. This process of alternating between `E` and `M` steps is repeated until convergence, i.e., until finding the model parameters that best fit the observed data and our prior belief.

During the test time, for any given unseen data point $\boldsymbol{x}_t$, the approximate posterior distribution over latent variables $q(\boldsymbol{z}_t \mid \boldsymbol{\beta}_t)$ is obtained by following only the `VB E-step` (model parameters are kept constant). As described earlier, if the latent variable represents a discrete topic or class label, then $q(\boldsymbol{z}_t \mid \boldsymbol{\beta}_t)$ is the posterior probability of class labels. If the latent variable is a low dimensional continuous vector, then parameters ($\boldsymbol{\beta}_t$) of this posterior distribution can be seen as compact representation (embedding) of the test data $\boldsymbol{x}_t$.

The following sections present some of the popular generative models for documents; most of them use the above described VB technique.

3.2 Classical topic models

Majority of the topic models (Deerwester et al., 1990; Blei et al., 2003; Blei and Lafferty, 2005; Blei, 2012; Zhu and Xing, 2011; Srivastava et al., 2013b; Miao et al., 2016) for learning document representations are built on "bag-of-words" simplification of documents. It is defined as follows:

From a collection of documents, a matrix $\boldsymbol{X}$ of dimension $D \times V$ is constructed, where every row index $d = 1 \dots D$ represents a document, every column index $i = 1 \dots V$ denotes a word from the vocabulary. The value x_{di} in each cell corresponds to the number of occurrences of word i in document d.

Since every word does not appear in every document, the matrix $\boldsymbol{X}$ is very sparse ($> 90\%$). Moreover, not all the documents are of equal length, hence the variance in word counts is very high. Additionally, function (stop) words appear more frequently as compared to content words. All these characteristics of "bag-of-words" model make it difficult to extract semantic information from the documents.

A common way to address this problem is to remove stop words[5] from the documents and/or apply term frequency-inverse document frequency (TF-IDF) weighting: The word counts in each document are normalized between $[0, 1]$ and words that appear in every document are de-weighted, whereas words that appear in fewer documents are given higher weights. The TF-IDF weighting is computed as follows:

$$\mathrm{tf}(i, d) = \frac{c_{di}}{\max\limits_{j} c_{dj}}, \tag{3.15}$$

$$\mathrm{idf}(i, D) = \ln\Big(\frac{D}{N_{di} + 1}\Big), \tag{3.16}$$

$$\mathrm{tfidf}(i, d, D) = \mathrm{tf}(i, d) \times \mathrm{idf}(i, D), \tag{3.17}$$

where c_{di} in (3.15) refers to the number of occurrences of word i in document d. N_{di} in (3.16) refers to the number of documents in which word i appears. The $+1$ in denominator of (3.16) avoids division by zero and is interpreted as: "each word from the vocabulary appears in at least one document". Often, in practice the following smoothed version of idf is used:

$$\mathrm{idf}(i, D) = 1 + \ln\Big(\frac{D + 1}{N_{di} + 1}\Big). \tag{3.18}$$

TF-IDF weighting does not necessarily bring out the semantic relations of words present in the documents, and moreover, the documents represented using TF-IDF are of very high-dimension i.e., equal to the size of the vocabulary.

[5] The list of stop words have to be constructed manually.

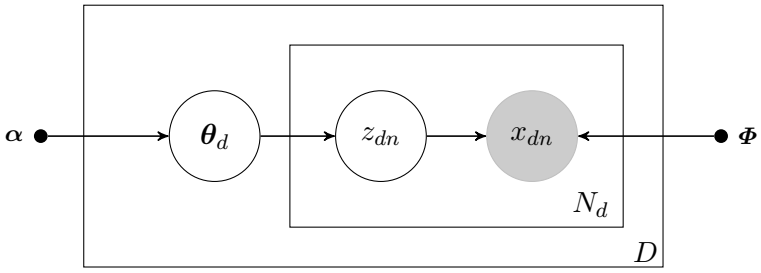


Figure 3.2: Graphical representation of latent Dirichlet allocation

3.2.1 Latent semantic analysis

Latent semantic analysis (LSA) (Deerwester et al., 1990) is arguably the origin for topic models. It is based on singular value decomposition of TF-IDF weighted counts matrix $\widehat{\boldsymbol{X}}$ (every element $\hat{x}_{ij}$ is computed according to (3.17)):

$$\widehat{\boldsymbol{X}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}, \tag{3.19}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}^{\mathsf{T}}$ are orthogonal matrices and the diagonal matrix $\boldsymbol{\Sigma}$ contains singular values of $\widehat{\boldsymbol{X}}$. The document and word representations are obtained by dimensionality reduction. This is done by considering only K largest singular values from $\boldsymbol{\Sigma}$, (where $K \ll V, K \ll D$) and setting the remaining values to zeros. If the resulting singular matrix is denoted by $\tilde{\boldsymbol{\Sigma}}$, then the rows of $\boldsymbol{U}\tilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}$ could be seen as document representations (co-ordinates for documents in latent space) and columns of $\tilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}\boldsymbol{V}^{\mathsf{T}}$ as word representations (co-ordinates for words) in the same latent space.

The original high-dimensional document vectors in $\boldsymbol{X}$ are sparse but the corresponding low dimensional vectors are not. This suggests that it is possible to associate documents meaningfully even if they do not share common words. Further, it also allows documents and words to be projected on to the same latent space. Document similarity can be computed using cosine or Euclidean distance among vectors in the latent space.

LSA inspired many other models, such as probabilistic latent semantic analysis (Hofmann, 1999) and latent Dirichlet allocation (Blei et al., 2003).

3.2.2 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is one of the most popular probabilistic topic models. LDA is seen as a generative model for bag-of-words representation of documents, with assumptions about a document as a mixture of latent topics and each topic as a mixture of words in the vocabulary. The generative process assumed by LDA is described as follows:

Let $\boldsymbol{\Phi}$ be a low-rank matrix of size $K \times V$ (where $K \ll V$), denote parameters of the model that represent a topic-word mixture. Every row $\boldsymbol{\varphi}_k \in \boldsymbol{\Phi}$ is a topic-specific discrete

probability distribution over the vocabulary of size V, i.e., $\sum_{i=1}^{V} \varphi_{ki} = 1$. This also means that $\boldsymbol{\varphi}_k$ lives on $V-1$ simplex ($\triangle^{\mathsf{V-1}}$). Any point on simplex represents a proper discrete probability distribution, i.e., the co-ordinates sum up to one.

Given the model parameters $\boldsymbol{\Phi}$, every document is assumed to be generated according to the following stochastic process:

First, a K dimensional document specific latent variable $\boldsymbol{\theta}_d$ is drawn from a Dirichlet distribution:

$$\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\theta}_d \,|\, \boldsymbol{\alpha}), \tag{3.20}$$

where $\boldsymbol{\alpha}$ is the concentration parameter of Dirichlet distribution.

This can also be seen as having a prior distribution over latent variables:

$$p(\boldsymbol{\theta}_d) = \mathrm{Dir}(\boldsymbol{\theta}_d \,|\, \boldsymbol{\alpha}), \tag{3.21}$$

$$= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1}, \tag{3.22}$$

where $\Gamma(\alpha) = (\alpha - 1)!$ represents Gamma function.

The document vectors $\boldsymbol{\theta}_d$ live on $\triangle^{\mathsf{K-1}}$. A few examples are shown in Fig. 3.3, where each sub figure shows samples drawn from Dirichlet distribution with a different concentration parameter $\boldsymbol{\alpha}$. All the samples live in 2-simplex ($\triangle^2$).

For each word position $n \; \forall \, n = 1 \ldots N_d$ in document d, a topic indicator variable is sampled:

$$z_{dn} \sim \mathrm{Multi}(\boldsymbol{\theta}_d, 1), \tag{3.23}$$

which is then used to sample a word token x_{dn} from the corresponding topic specific distribution:

$$x_{dn} \sim \mathrm{Multi}(\boldsymbol{\varphi}_{z_{dn}}, 1). \tag{3.24}$$

The multinomial distribution with one trial is also known as Categorical distribution.

In LDA, the topic ($\boldsymbol{\varphi}_k$) and document ($\boldsymbol{\theta}_d$) vectors live in $(V-1)$ and $(K-1)$ simplexes respectively. Every word x_{dn} in a document d is associated with a discrete latent variable z_{dn} that tells which topic was responsible for generating the word. This can be seen from the corresponding graphical model in Fig. 3.2. From the graphical model, we can also see that the

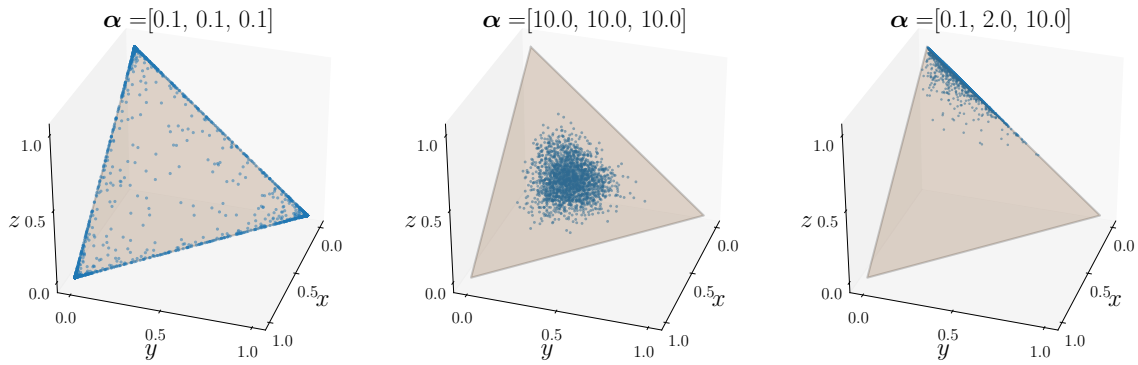


Figure 3.3: Samples from Dirichlet distribution (points in 2-simplex $\triangle^2$).

joint distribution of all documents (matrix of word counts $\boldsymbol{X}$) and the corresponding latent variables $\boldsymbol{\Theta}, \boldsymbol{Z}$ factorizes:

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta} \,|\, \boldsymbol{\Phi}, \boldsymbol{\alpha}) = \prod_{d=1}^{D} p(\boldsymbol{x}_d, \boldsymbol{z}_d, \boldsymbol{\theta}_d \,|\, \boldsymbol{\Phi}, \boldsymbol{\alpha}) \tag{3.25}$$

$$= \prod_{d=1}^{D} p(\boldsymbol{\theta}_d \,|\, \boldsymbol{\alpha})\, p(\boldsymbol{x}_d, \boldsymbol{z}_d \,|\, \boldsymbol{\theta}_d, \boldsymbol{\Phi}) \tag{3.26}$$

$$= \prod_{d=1}^{D} p(\boldsymbol{\theta}_d \,|\, \boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(z_{dn} \mid \boldsymbol{\theta}_d)\, p(x_{dn} \mid z_{dn}, \boldsymbol{\Phi}) \tag{3.27}$$

3.2.2.1 Inference in LDA

During inference, the generative process is inverted to obtain posterior distribution over latent variables, $p(\boldsymbol{\theta}_d, \boldsymbol{z}_d \,|\, \boldsymbol{x}_d, \boldsymbol{\alpha}, \boldsymbol{\Phi})$, given the observed data and the prior belief. This can be written using Bayes' rule[6]:

$$p(\boldsymbol{\theta}, \boldsymbol{z} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Phi}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi})}{p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi})} \tag{3.28}$$

The denominator in the above equation is marginal of observed data and is obtained by integrating over $\boldsymbol{\theta}$ and summing over $\boldsymbol{z}$:

$$p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}) = \int p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \left(\prod_{n} \sum_{z_n} p(z_n \mid \boldsymbol{\theta}) p(x_n \mid z_n, \boldsymbol{\Phi}) \right) \mathrm{d}\boldsymbol{\theta} \tag{3.29}$$

$$= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \int \left(\prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \right) \left(\prod_{n} \sum_{k=1}^{K} \prod_{j=1}^{V} (\theta_k \Phi_{kj})^{x_{nj}} \right) \mathrm{d}\boldsymbol{\theta} \tag{3.30}$$

[6] Omitting the document suffix d for clarity of presentation.

The integral in the above equation is intractable because of the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$. To resolve it, (Blei et al., 2003) resorted to variational inference that finds an approximation to the true posterior with a variational distribution $q(\boldsymbol{\theta}_d, \boldsymbol{z}_d)$. Further the following approximation was made, to make the inference tractable:

$$q(\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) \prod_n q(z_n \mid \phi_n), \tag{3.31}$$

where $\boldsymbol{\gamma}$ and ϕ represent Dirichlet and multinomial parameters of variational distribution respectively. Using the VB formulation (3.14) the log marginal for LDA is given by:

$$\ln p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}) = \mathcal{L}(q(\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{\gamma}, \boldsymbol{\phi})) + D_{\mathrm{KL}}(q(\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{\gamma}, \boldsymbol{\phi}) \,||\, p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi})) \tag{3.32}$$

Expanding $\mathcal{L}(q)$ using (3.13), (3.27) and (3.31):

$$\begin{aligned} \mathcal{L}(q) = {} & \mathbb{E}_q[\ln p(\boldsymbol{\theta} \mid \boldsymbol{\alpha})] + \mathbb{E}_q[\ln p(\boldsymbol{z} \mid \boldsymbol{\theta})] + \mathbb{E}_q[\ln p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\Phi})] \\ & - \mathbb{E}_q[\ln q(\boldsymbol{\theta})] - \mathbb{E}_q[\ln q(\boldsymbol{z})] \end{aligned} \tag{3.33}$$

The complete expansion of ELBO from (3.33), can be found in (Blei et al., 2003) Next, following the VB training procedure (Algorithm 1), one can obtain the model parameters and variational distribution. The complete derivation of update formulae are given in (Blei et al., 2003). The parameters $\boldsymbol{\Phi}$ were obtained using maximum-likelihood approach.

3.2.2.2 Limitations

There are two problems with the assumptions made by LDA:

1. The first one is the choice of Dirichlet-multinomial over document-topic mixture. Although it simplifies the inference process because of the conjugacy; the assumption of Dirichlet distribution causes limitations to the model, and $q(\boldsymbol{\theta}_d)$ cannot capture the correlations (Blei and Lafferty, 2005) between topics in each document, i.e., every document contributes to every latent topic with a non-zero probability. For example, consider a huge collection of documents (archive of news articles) with topics such as **health**, **diseases**, **automotive**, **sports**, **space**, **pc-hardware** and so on. In such a collection, it is reasonable to assume that a subset of latent topics are highly correlated (e.g.**health** and **diseases**). Similarly there exists topics that are completely unrelated (e.g. **diseases**, and **pc-hardware**). If a document belongs to a topic of **health**, then according to LDA, it also belongs to a latent topics corresponding to **diseases**, **pc-hardware**, **sports**, **space**, and **automotive** with a non-zero probability. This is because of the Dirichlet assumption of topic proportions over documents (Fig: 3.3). In reality, it is highly unlikely (or impossible) for a document to belong all the topics. The Dirichlet distribution cannot capture any negative correlations.

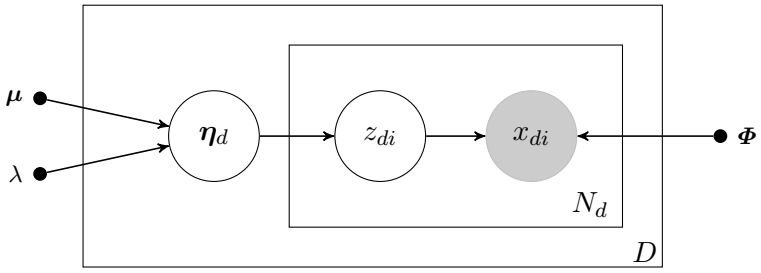


Figure 3.4: Graphical representation of correlated topic model.

2. The second assumption LDA makes is that every (latent) topic vector $\boldsymbol{\varphi}_k$ is a discrete probability distribution over all the words in the vocabulary, i.e., every word contributes to every topic with a non-zero probability. This may not be a reasonable assumption. For example consider a set of words (names of viruses) like *plasmodium, falciparum, malariae, ovale, vivax*[7], which are high correlated with topics such as **health** and **diseases**. Next, consider another set of words such as *microarchitecture, cache, pentium* which are highly correlated with topics such as **pc-hardware** and **technology**. These two sets of words are highly un-correlated, i.e., the presence of one set of words in a document implies that the other set of words cannot appear (a strong negative correlation). Moreover, the names of viruses do not contribute to the topics related to **pc-hardware** and **technology**. But according to LDA, every word belongs to every topic with a non-zero probability.

To overcome the first limitation Blei and Lafferty (2005) proposed to model document vectors ($\boldsymbol{\theta}$) with Gaussian distribution, and the resulting model is called correlated topic model (CTM).

3.2.3 Correlated topic model

The generative process of CTM is the same as in LDA except for document vectors are drawn from Gaussian instead of Multinomial ((3.21) is replaced by the following):

$$p(\boldsymbol{\eta}_d) = \mathcal{N}(\boldsymbol{\eta}_d \mid \boldsymbol{\mu}, (\lambda \boldsymbol{I})^{-1}), \tag{3.34}$$

$$\boldsymbol{\theta}_d = \text{softmax}(\boldsymbol{\eta}_d). \tag{3.35}$$

In this formulation, the document vectors $\boldsymbol{\eta}_d$ are no longer in the $(K-1)$ simplex, rather they are dependent through the logistic Normal. Fig. 3.5 shows samples from logistic Normal. The advantage is that the documents vectors can model the correlations in topics. The topic distributions over vocabulary $\boldsymbol{\Phi}$, however still remained discrete.

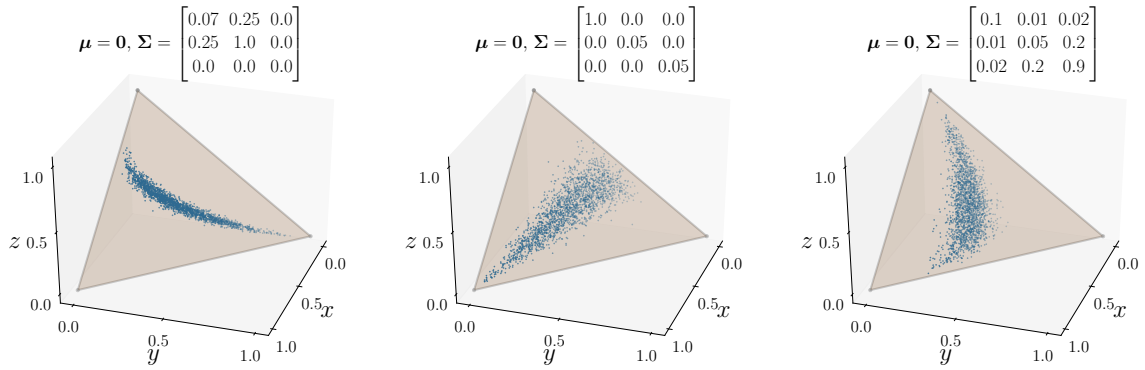


Figure 3.5: Samples from Logistic Normal.

3.2.3.1 Inference in CTM

The true posterior in CTM is intractable, which can be seen by examining the following equation:

$$p(\boldsymbol{\eta}, \boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\mu}, \lambda, \boldsymbol{\Phi}) = \frac{p(\boldsymbol{\eta} \mid \boldsymbol{\mu}, \lambda) \prod_i p(z_i \mid \boldsymbol{\eta}) p(x_i \mid z_i, \boldsymbol{\Phi})}{\int p(\boldsymbol{\eta} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_i \sum_{z_i=1}^{K} p(z_i \mid \boldsymbol{\eta}) p(x_i \mid z_i, \boldsymbol{\Phi}) \, \mathrm{d}\boldsymbol{\eta}} \tag{3.36}$$

The denominator in above equation is intractable for two reasons: (i) the sum over K values of z_i occurs inside the product which results in combinatorial number of terms, (ii) the distribution of topic proportions $p(\boldsymbol{\eta} \mid \boldsymbol{\mu}, \lambda)$ is not conjugate to $p(z_i \mid \boldsymbol{\eta})$ and integral cannot be computed analytically.

Blei and Lafferty (2005) resorted to variational inference to find an approximation to the true posterior with a variational distribution $q(\boldsymbol{\eta}, \boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi})$. Further mean-field approximation was used to make the inference tractable (same as in LDA (3.31)):

$$q(\boldsymbol{\eta}, \boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\eta} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}) \prod_i q(z_i \mid \phi_i), \tag{3.37}$$

where $\boldsymbol{\nu}, \boldsymbol{\gamma}$ are mean and precision of Gaussian distribution and $\boldsymbol{\phi}$ are the multinomial parameters. Following the standard VB approach, we can write log marginal as:

$$\ln p(\boldsymbol{x} \mid \boldsymbol{\mu}, \lambda, \boldsymbol{\Phi}) = \mathcal{L}(q(\boldsymbol{\eta}, \boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi})) + D_{\mathrm{KL}}(q(\boldsymbol{\eta}, \boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi}) \,||\, p(\boldsymbol{\eta}, \boldsymbol{z}, \boldsymbol{x} \mid \boldsymbol{\mu}, \lambda, \boldsymbol{\Phi})) \tag{3.38}$$

Expanding $\mathcal{L}(q)$ by using the factorization:

$$\mathcal{L}(q) = \mathbb{E}_q[\ln p(\boldsymbol{\eta} \mid \boldsymbol{\mu}, \lambda)] + \sum_i \underbrace{\mathbb{E}_q[\ln p(z_i \mid \boldsymbol{\eta})]}_{\mathsf{A}} + \sum_i \mathbb{E}_q[\ln p(x_i \mid z_i, \boldsymbol{\Phi})] + \mathrm{H}[q], \tag{3.39}$$

where $\mathrm{H}[q]$ is the entropy of variational distribution. The term A in the (3.39) is intractable as it involves solving the expectation over log-sum-exp function:

$$\mathbb{E}_q[\ln p(z_i \mid \boldsymbol{\eta})] = \mathbb{E}_q[\boldsymbol{\eta}_{z_i}^{\mathsf{T}}] - \mathbb{E}_q\left[\ln\left(\sum_{k=1}^{K} \exp\{\eta_k\}\right)\right] \tag{3.40}$$

[7] https://www.who.int/ith/diseases/malaria/en/

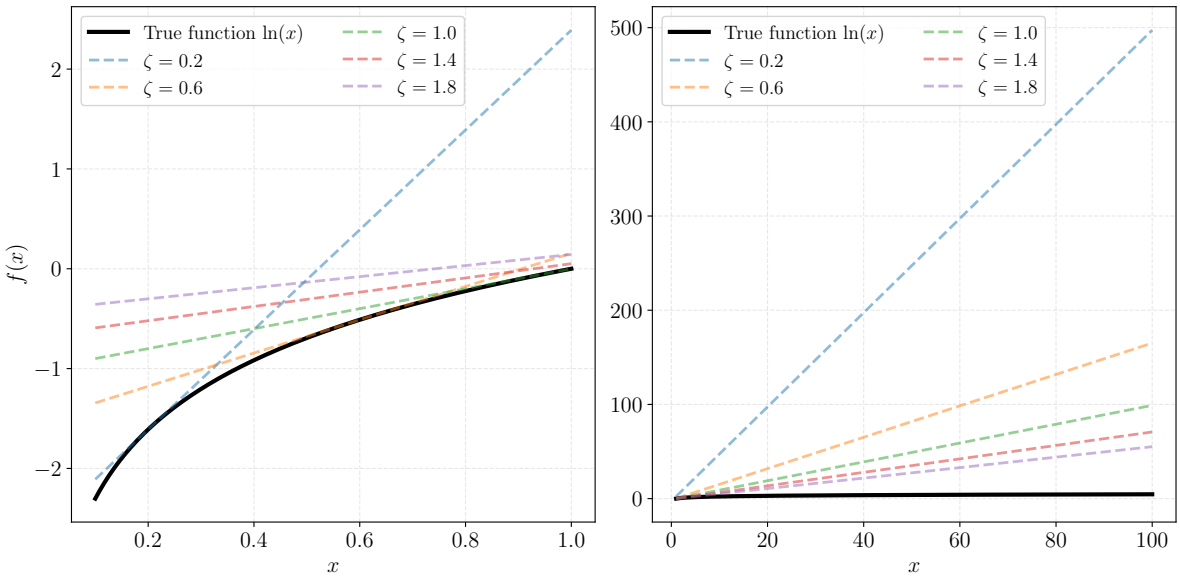


Figure 3.6: First-order Taylor series approximation of $\ln(x)$ at $\zeta \in [0.2, 2.0]$. The approximation provides a tighter bound for $0 < x < 1$ and loose bounds for $x > 1$.

This intractability (expectation over log-sum-exp or log normalizer) is a generic problem that arises while performing variational inference in mixed-logit models (Depraetere and Vandebroek, 2017). In CTM, the authors used *first-order* Taylor series expansion[8] to form an upper bound on the negative log normalizer:

$$\mathbb{E}_q\left[\ln\left(\sum_{k=1}^{K}\exp\{\eta_k\}\right)\right] \geq \mathbb{E}_q\left[\ln(\zeta) + \frac{1}{\zeta}\Big(\sum_{k=1}^{K}\exp\{\eta_k\} - 1\Big)\right] \tag{3.41}$$

$$= \ln(\zeta) + \zeta^{-1}\left(\sum_{k=1}^{K}\mathbb{E}_q[\exp\{\eta_k\}] - 1\right), \tag{3.42}$$

where ζ is an additional variational parameter. The remaining terms in (3.39) have analytical form. See Appendix of (Blei and Lafferty, 2007) for the step-by-step derivation of each term. Finally, the objective to optimize is a lower bound on ELBO (because of the upper bound used in (3.42)). CTM is trained by using the standard `VB EM` algorithm.

The example given in Fig. 3.6 illustrates the problems of using first-order Taylor series approximation of $\ln(x)$. Notice that for $0 < x < 1$, the approximations provide a tighter bound to the true function (left sub-plot), but for $x >> 1$, the approximation results in a loose bound to the true function (right sub-plot), which can cause instabilities during training.

In this thesis, the same problem of intractability is encountered while performing variational

[8] Taylor series expansion of an infinitely differentiable function $f(x)$ at a point ζ is $f(\zeta) + \frac{f'(\zeta)}{1!}(x-\zeta) + \frac{f''(\zeta)}{2!}(x-\zeta)^2 \ldots$

inference in the proposed Bayesian SMM; which is resolved using Monte Carlo approximation via re-parametrization trick (Kingma and Welling, 2014). The variational inference for the proposed Bayesian SMM is derived and explained in Chapter 5.

CTM was applied to the archives of *Science* present in JSTOR[9], and was shown to give a better fit to the data as compared to LDA. Although the correlations among topic-document mixture are captured by Gaussian distribution, CTM still constraints the topic-word distributions to be discrete. It means that every word belongs to every latent topic with a non-zero probability (limitation 2 from § 3.2.2.2). To address this, one can introduce Laplace priors or apply ℓ_1 regularization over topic-word mixture (Shashanka et al., 2007); which can introduce explicit zeros into the topic-word mixture. The proposed model in this thesis, ℓ_1 SMM makes use of ℓ_1 regularization over model parameters (topic-word mixture) and thus avoids the shortcomings of LDA and CTM. More details with experimental evidence is presented in Chapter 4.

3.3 Sparse topic models

Sparsity is often one of the desired properties (Eisenstein et al., 2011; Shashanka et al., 2007) in topic models. It is difficult to introduce sparsity into Dirichlet-multinomial mixture based topic models such as LDA or CTM (Ganchev et al., 2009). Sparse coding inspired topic model, sparse topical coding (STC) was proposed by (Zhu and Xing, 2011), where the authors have obtained sparse representations for both documents and words.

3.3.1 Sparse topical coding

Sparse coding aims to learn a set of basis (or atoms in dictionary) and *sparse codes* (representations) of input data in a way that their linear combination reconstructs the original input data (eg. word counts in a document). Thus sparse coding can be viewed as a constrained optimization problem.

A graphical representation of sparse topical coding model is presented in Fig. 3.7, where the vector $\boldsymbol{\varphi}_k$ represents a topic basis i.e., a uni-gram distribution over vocabulary of size V. Let $\boldsymbol{\Phi}_{K\times V}$ represent the dictionary of K such topic bases. $\boldsymbol{\theta}_d$ represents a document specific code (representation). Each observed word count x_{di} in document d is assumed to be generated by the following two steps:

1. sampling a word code $\boldsymbol{s}_{di}$ from conditional distribution $p(\boldsymbol{s}_{di} \mid \boldsymbol{\theta}_d)$:

$$p(\boldsymbol{s}_{di} \mid \boldsymbol{\theta}_d) \propto \exp\{-\gamma||\boldsymbol{s}_{di} - \boldsymbol{\theta}_d||_2 - \rho||\boldsymbol{s}_{di}||_1\} \tag{3.43}$$

2. sampling word count x_{di} from a Poisson distribution with mean parameter $\nu_{di} = \boldsymbol{s}_{di}^{\mathsf{T}}\boldsymbol{\varphi}_{\cdot i}$:

[9] https://www.jstor.org/

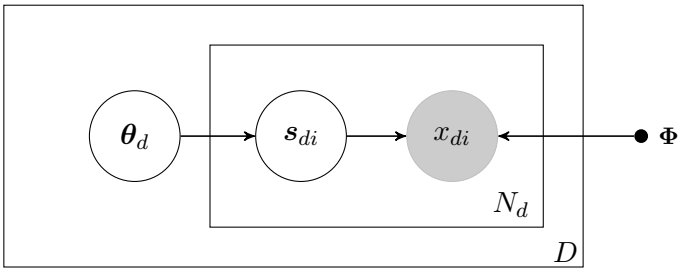


Figure 3.7: Representation of sparse topical coding (STC) model.

$$p(x_{di} \mid \nu_{di}) = \text{Poisson}(x_{di}; \nu_{di}) \tag{3.44}$$

$$= \frac{\nu_{di}^{x_{di}} \exp\{-\nu_{di}\}}{x_{di}!}, \tag{3.45}$$

where $\boldsymbol{\varphi}_{\cdot i}$ represents a column in dictionary $\boldsymbol{\Phi}$.

Given the observed data $\boldsymbol{X}_{D \times V}$ i.e, bag-of-words, STC aims to find the point estimates of codes $\Theta = \{\boldsymbol{\theta}_d, \boldsymbol{s}_d\}_{d=1}^{D}$ and the dictionary $\boldsymbol{\Phi}$, by minimizing the following constrained objective:

$$f(\Theta, \boldsymbol{\Phi}) = \underset{\Theta, \boldsymbol{\Phi}}{\text{minimize}} \sum_{d=1}^{D} \sum_{i=1}^{N_d} \Big[\ln \text{Poisson}(x_{di}; \nu_{di}) + (\gamma ||\boldsymbol{s}_{di} - \boldsymbol{\theta}_d||_2 - \rho ||\boldsymbol{s}_{di}||_1) \Big] + \lambda ||\boldsymbol{\theta}_d||_1 \tag{3.46}$$

$$\text{subject to:} \quad \boldsymbol{\theta}_d \geq 0 \quad \forall d\,; \tag{3.47}$$

$$\boldsymbol{s}_{di} \geq 0 \quad \forall d, i\,; \tag{3.48}$$

$$\sum_{i=1}^{V} \varphi_{ki} = 1 \quad \forall k = 1 \ldots K\,, \tag{3.49}$$

where ℓ_1 regularization is applied over document codes $\boldsymbol{\theta}_d$ and word codes $\boldsymbol{s}_{di}$.

3.3.1.1 Optimization

The objective function $f(\Theta, \boldsymbol{\Phi})$ from (3.46) is minimized by following co-ordinate descent algorithm. More specifically, the procedure alternately performs descent w.r.t $\Theta = \{\boldsymbol{s}_d, \boldsymbol{\theta}_d\}_{d=1}^{D}$, with a fixed $\boldsymbol{\Phi}$; and then w.r.t $\boldsymbol{\Phi}$ with fixed Θ. Detailed steps are given in (Zhu and Xing, 2011).

Zhu and Xing (2011) has shown that STC learns sparse representation of document codes and also performs better than LDA at document classification task on *20Newsgroups* corpus. Further, it was shown that STC can be trained in a discriminative fashion by jointly optimizing a convex combination of original objective in (3.46) and hinge loss (objective of support vector machines).

Although STC achieves sparsity and performs better than LDA, it still cannot model the correlations present between latent topics-words and document-topics. This can be seen by examining the constraints used in optimizing the STC objective. The dictionary $\boldsymbol{\Phi}$, which is

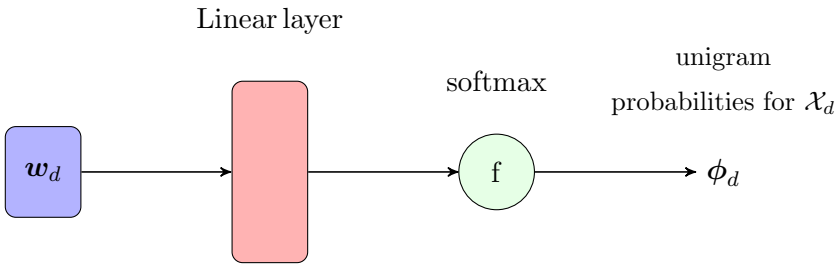


Figure 3.8: Paragraph vector: distributed bag-of-words (PV-DBOW) model. The document or paragraph-specific embedding $\boldsymbol{z}_d$ is stochastically trained to maximize the probabilities ($\boldsymbol{\phi}_d$) of a subset of words ($\mathcal{X}_d$) present in document d.

equivalent to topic-word mixture in LDA and CTM is forced to live in a simplex i.e., every basis contributes to every latent topic with non-zero probability (limitation 2 from § 3.2.2.2). Next, the document and word codes are constrained to be non-negative. Although non-contributing latent topics can be set to zeros, it cannot model strong negative correlations.

3.4 Neural network based topic models

The advances in neural networks and deep learning have lead to the development of several models in the NLP community and also advanced state-of-the-art in several tasks. Two important topic models are worth mentioning as they are related to the models proposed in this thesis. The first one is "bag-of-words" variant of paragraph vector (PV-DBOW) (Le and Mikolov, 2014) and the second one is neural variational document model (NVDM) (Miao et al., 2016), an adaptation of variational auto encoders (Kingma and Welling, 2014) for document modelling.

3.4.1 Paragraph vector

Inspired by the popular `word2vec` (Mikolov et al., 2013) model, paragraph vector aims to learn compact embeddings of sentences or paragraphs (or documents). Le and Mikolov (2014) proposed two models, and one of them is closely related to modelling bag-of-words. This model termed as PV-DBOW is shown in Fig. 3.8, where $\boldsymbol{z}_d$ is a compact representation of a document that aims to predict a set of words ($\mathcal{X}_d$) from a document. In every step of training, the set of words $\mathcal{X}_d$ are randomly sampled from a document d, and the parameters of the model are updated via stochastic gradient descent, where the gradient is computed for the negative

log-likelihood of the data:

$$\mathcal{L} = -\sum_{\forall d} \sum_{\forall \mathsf{x}_i \in \mathcal{X}_d} \ln p(\mathsf{x}_i \mid \phi_{di}), \tag{3.50}$$

$$\phi_{di} = \frac{\exp\{\boldsymbol{w}_i^\top \boldsymbol{z}_d + b_i\}}{\sum_{\forall \mathsf{x}_j \in \mathcal{S}_d} \exp\{\boldsymbol{w}_j^\top \boldsymbol{z}_d + b_j\}}, \tag{3.51}$$

where $\mathcal{S}_d$ are set of words that are not present in document d; also called as negative samples. During inference, only the document embedding $\boldsymbol{z}_d$ is updated by keeping the rest of the parameters fixed. This model bears similarities with subspace multinomial model (SMM) (Kockmann et al., 2010) as we will see in Chapter 4.

3.4.2 Neural variational document model

Neural variational document model (NVDM) (Miao et al., 2016) is an adaptation of variational autoencoder (VAE) (Kingma and Welling, 2014) for modelling bag-of-words representation of documents. A standard autoencoder has two parts; the first one called the encoder (multi-layered neural network) learns compact (latent) representation $\boldsymbol{z}$ of input data $\boldsymbol{x}$; the later part called decoder (multi-layered neural network) aims to reconstruct the input data $\hat{\boldsymbol{x}}$ from the latent representation. Usually, the autoencoders are trained to minimize mean-squared error between the input $\boldsymbol{x}$ and the reconstructed output $\hat{\boldsymbol{x}}$, by updating the weights of neural network via back-propagation.

In VAE, the encoder attempts to learn the (posterior) distribution of latent variables $p(\boldsymbol{z} \mid \boldsymbol{x})$ given the input data $\boldsymbol{x}$; the decoder aims to model the data, given the latent variable $p(\boldsymbol{x} \mid \boldsymbol{z})$. VAE learns to approximate the true posterior by variational distribution $q(\boldsymbol{z} \mid \boldsymbol{x})$. Moreover, if the functional forms of posterior and data (likelihood) are not conjugate to each other, the posterior becomes intractable. This problem is similar to what we have seen so far in LDA and CTM. In the original VAE, the authors (Kingma and Welling, 2014) proposed to approximate the intractable functions using Monte Carlo samples via the re-parametrization trick. We will revisit this problem in detail when we present our Bayesian model in Chapter 5.

In NVDM, the encoder parts takes every document (vector of word counts) $\boldsymbol{x}_d$ as input and predicts posterior distribution of latent variables $p(\boldsymbol{z}_d \mid \boldsymbol{x}_d)$. The decoder takes a latent variable and predicts the parameters $\boldsymbol{\theta}_d$ of multinomial distribution that can model the distribution of data $p(\boldsymbol{x}_d \mid \boldsymbol{\theta}_d, \boldsymbol{z}_d)$. The model is trained by stochastic gradient descent that learns weights of the neural network (encoder and decoder) and also the parameters of the posterior distribution. During test time, given a document, the posterior distribution is obtained just by forward propagating through the encoder part. However, this process of obtaining the posterior distribution from the encoder is sub-optimal, as we will show in Chapter 5.

The authors have shown that NVDM achieves state-of-the-art perplexity results on *20Newsgroups* text corpus (Miao et al., 2016). Chapter 5 presents Bayesian SMM that shares similari-

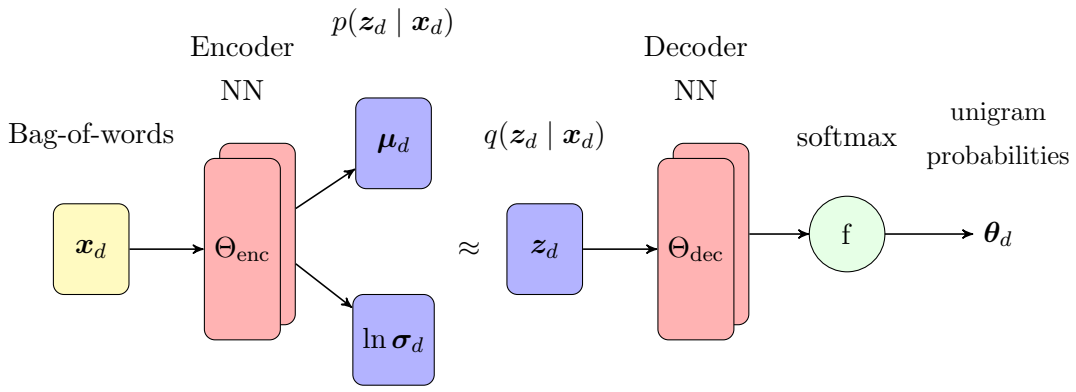


Figure 3.9: Neural variational document model. Left part is the encoder predicting the parameters of the posterior distribution of latent variables $p(\boldsymbol{z}_d \mid \boldsymbol{x}_d, \Theta_{\text{enc}})$. The right part is the decoder that generates parameters $(\boldsymbol{\theta}_d)$ of the document-specific uni-gram distribution over vocabulary $p(\boldsymbol{x}_d \mid \boldsymbol{z}_d, \Theta_{\text{dec}})$.

ties with NVDM, as both of them maximize expected log-likelihood of data (bag-of-words representation of documents), assuming multinomial distribution. The experiments show that Bayesian SMM achieves superior perplexity scores on *20Newsgroups* data and outperforms NVDM with a significant margin.

3.4.3 Sparse composite document vector

Mekala et al. (2017) proposed an algorithm to obtain sparse document embeddings called sparse composite document vector (SCDV) from pre-trained word embeddings. It was shown to achieve superior classification results on *20Newsgroups* corpus as compared to paragraph vector, neural tensor skip-gram model (Liu et al., 2015). The experiments in Chapter 6 shows that the proposed model achieves comparable (slightly superior) classification results to SCDV.

3.4.4 Discriminative text classifiers

The other category of neural network based models are text classifiers. Unlike topic models, they are not generative models for documents, but are discriminative models for document / text classification. Recent works include character level convolutional neural networks (Zhang et al., 2015), and hierarchical attention based networks (Yang et al., 2016b) for document classification. The major limitations of discriminative models is their dependency on labelled data and the difficulty in adapting to newer classes or domains.

3.4.5 Pre-trained language models

These models are trained on largely available unlabelled data using self-supervised objectives (Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019), such as next-word prediction or masked word-prediction. Once the model is trained on the word-prediction task, it then fine-tuned on a particular dataset for a specific task such as document classification or natural language inference. Hence the name 'pre-training'. The models primarily employ several layers of BiLSTMs or transformer blocks (Vaswani et al., 2017), and comprise of millions of parameters. The 'pre-training' step is computationally intensive and usually takes few days on several GPUs (Radford, 2018; Devlin et al., 2019) or TPUs (Yang et al., 2019). Although, the fine-tuning step is faster as compared to the original pre-training, it requires to load the entire model into the memory to optimize its parameters.

3.5 Summary and relation to the work in this thesis

This chapter reviewed the concept of generative models, and variational inference approach, followed by a detailed overview of most popular and recent topic models with their assumptions and consequent limitations. Given this background on topic models, the relation to work in this thesis is outlined below:

This thesis proposes (Kesiraju et al., 2016) the use of subspace multinomial model for learning document representations[10]. The relation between SMM and paragraph vector (PV-DBOW) will be shown. Further, a novel variant of SMM that is based on sparse (ℓ_1) regularization of parameters of SMM will be presented. We have seen that sparsity helps in text modelling, especially parameters corresponding to topic-word mixtures can benefit from having sparse priors (Zhu and Xing, 2011). Consequently, ℓ_1 regularization makes the learning difficult as the objective function becomes non-differentiable. This problem is addressed by employing orthant-wise[11] learning (Andrew and Gao, 2007). The experiments on *20Newsgroups* corpus show that the proposed ℓ_1 SMM is superior to other unsupervised topic models such as LDA and STC on topic identification and document clustering tasks. Details of proposed ℓ_1 SMM are given in Chapter 4.

Next, a Bayesian modelling for SMM is presented; that results in a complete generative model for "bag-of-words" representation of documents. Using this Bayesian SMM, we are able to learn document representations along with their uncertainties. However, the Bayesian inference in SMM comes with the problem of intractability (solving expectation over log-sum-exp function); similar to the one seen in CTM. This problem is addressed by approximating the intractable function with Monte Carlo samples via re-parametrization trick. The experimental details show

[10] May et al. (2015) also worked on the same idea independently.

[11] Orthant is a generalization of quadrant to n-dimensional space.

that Bayesian SMM achieve state-of-the-art perplexity results on *20Newsgroups* text and *Fisher* speech corpus. Details are given in Chapter 5. Next, a generative classifier for the task of topic identification is presented, that exploits the uncertainty in posterior distribution of document representations.

Chapter 7 presents a multilingual extension of Bayesian SMM. The idea is to capture only semantics (topic-like) from document and suppress the language information. Such a model is especially useful in cross-lingual transfer applications where training data is scarce. The experiments on *Europarl* and *Reuters multi-lingual news* corpora show that the proposed multilingual Bayesian SMM is superior to multilingual word embeddings and neural machine translation inspired sequence-to-sequence BiLSTM models in zero-shot cross-lingual topic identification.

Finally, the thesis discusses (i) discriminative and hybrid variants of SMM that can exploit both the labelled and unlabelled data, thus making the best use of generative and discriminative approaches, and (ii) variants of SMM that exploit contextual n-gram information for learning sentence representations.

Chapter 4

Learning document representations using subspace multinomial model

This chapter presents the application of subspace multinomial model (SMM) to learn document embeddings[1]. These document embeddings have Gaussian-like distribution, which makes them compatible with simple generative classifiers such as Gaussian linear classifier. Additionally, the Gaussian nature of document embeddings enables us to use simple clustering algorithms such as k-means.

A novel variant of SMM called ℓ_1 SMM, that is based on ℓ_1 regularization of its parameters is also presented. The ℓ_1 regularization introduces sparsity into the model, which is one of the desired properties while modelling text documents (Zhu and Xing, 2011; Shashanka et al., 2007; Eisenstein et al., 2011; Mekala et al., 2017). Using the proposed ℓ_1 SMM, I show that the extracted document embeddings together with simple Gaussian linear classifier (GLC) achieve superior classification accuracy on topic identification from *20 Newsgroups* text corpora, when compared to systems based unsupervised topic models such as latent Dirichlet allocation and sparse topical coding.

SMM was originally proposed for modelling discrete prosodic features for the task of speaker verification (Kockmann et al., 2010). Later, SMM and its variant, subspace n-gram model (SnGM) were used for phonotactic language recognition (Soufifar et al., 2011, 2013). Similar model was proposed by (Maas et al., 2011), that was used for learning word vectors for sentiment analysis.

4.1 Subspace multinomial model

Like majority of the probabilistic topic models (Blei, 2012), SMM also models bag-of-words representation of a document (vector of word counts $\boldsymbol{x}_d$) by a multinomial distribution. Let V represent the vocabulary size, and $\boldsymbol{\theta}_d \in \triangle^{\mathsf{V}-1}$ represents the document specific parameters of multinomial distribution i.e., unigram probabilities of individual words in a document,

[1]The embeddings extracted from SMM are also referred to as document i-vectors.

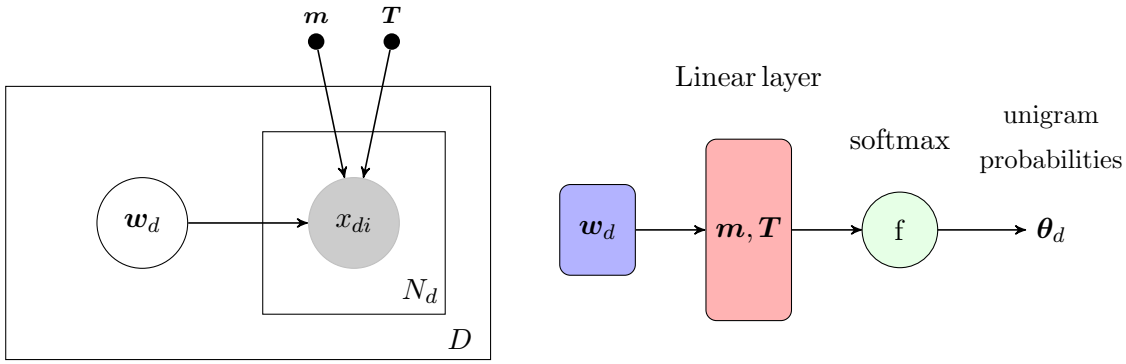


Figure 4.1: Graphical representation of SMM on the left, and alternative representation on the right. $\boldsymbol{w}_d$ is the document embedding, $\{\boldsymbol{m}, \boldsymbol{T}\}$ are the bias and weights of the linear layer.

then:

$$\boldsymbol{x}_d \sim \text{Multi}(\boldsymbol{\theta}_d, N_d), \tag{4.1}$$

where N_d denotes the number of tokens in document d.

SMM assumes that the document specific multinomial parameters $\boldsymbol{\theta}_d$ live in much smaller subspace defined as:

$$\boldsymbol{\theta}_d = \text{softmax}(\boldsymbol{m} + \boldsymbol{T}\,\boldsymbol{w}_d), \tag{4.2}$$

where $\{\boldsymbol{m}, \boldsymbol{T}\}$ are parameters of SMM and $\boldsymbol{w}_d$ is a document-specific latent variable. $\boldsymbol{T} \in \mathbb{R}^{V \times K}$ is a low-rank (total variability or bases) matrix that spans a K-dimensional subspace ($K \ll V$), and the vector $\boldsymbol{m} \in \mathbb{R}^V$ is bias or offset (also known as universal background model). The K dimensional latent variable $\boldsymbol{w}_d$ is seen as a low-dimensional representation for document $\boldsymbol{x}_d$ and is referred document embedding.

The graphical representation of SMM is depicted in Fig. 4.1; it bears some similarities with LDA and CTM. The major difference to note is the absence of discrete latent variable $\boldsymbol{z}$ in SMM. The alternative representation of SMM can be compared with paragraph vector-bag-of-words (PV-DBOW) from Fig. 3.8.

Given training documents $\boldsymbol{X} \in \mathbb{Z}^{*D \times V}$, SMM is trained by estimating its parameters $\{\boldsymbol{m}, \boldsymbol{T}\}$ that maximize the log-likelihood of the data, and for any given (unseen) test document $\boldsymbol{x}_t$, the corresponding latent representation (document embedding) $\boldsymbol{w}_t$ can be extracted.

The complete log-likelihood of the data $\boldsymbol{X}$ is the summation of log-likelihoods of individual documents $\boldsymbol{x}_d$. This can be seen from the graphical model in Fig. 4.1. According to the model, every document is a sample from multinomial distribution (4.1), hence the complete

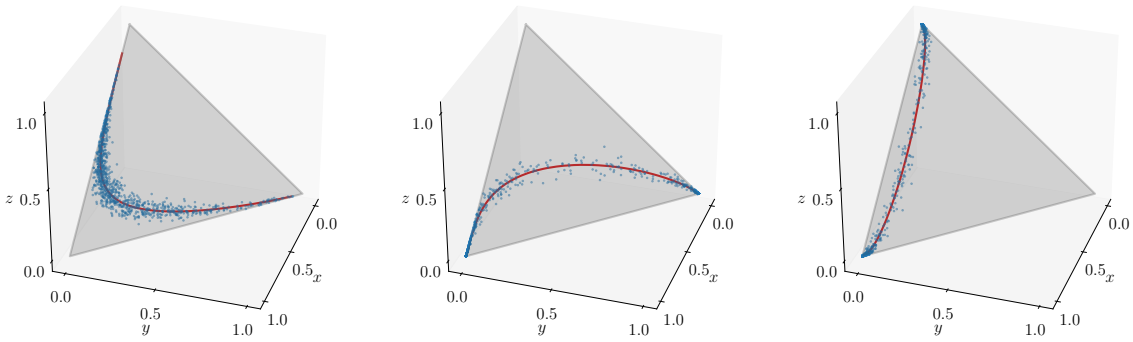


Figure 4.2: Illustration of one-dimensional subspace in 2-simplex. Every dot represents a sample (document).

log-likelihood is computed according to:

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\Theta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{x}_d \mid \boldsymbol{\theta}_d), \tag{4.3}$$

$$= \sum_{d=1}^{D} \sum_{i=1}^{V} x_{di} \ln(\theta_{di}), \tag{4.4}$$

$$= \sum_{d=1}^{D} \sum_{i=1}^{V} x_{di} \ln \left(\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}}{\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}} \right) \tag{4.5}$$

where $\boldsymbol{t}_i$ corresponds to a row in the matrix $\boldsymbol{T}$.

SMM with one-dimensional subspace is illustrated in Fig. 4.2.

4.1.1 Training

Following the similar training procedure from (Kockmann et al., 2010), the universal background model $\boldsymbol{m}$ is initialized with document-independent log uni-gram probabilities estimated from the training data $\boldsymbol{X}$ as:

$$m_i = \ln \left(\frac{\sum_d x_{di}}{\sum_i \sum_d x_{di}} \right) \quad \forall\, i = 1 \ldots V \tag{4.6}$$

The model is trained by alternating between the iterative updates of $\boldsymbol{T}$ and all the document embeddings $\boldsymbol{W}$. These updates are performed by following Newton-Raphson like update steps (Kockmann et al., 2010):

$$\boldsymbol{w}_d \leftarrow \boldsymbol{w}_d + \boldsymbol{H}_d^{-1} \nabla_{\boldsymbol{w}_d} \mathcal{L}\,, \tag{4.7}$$

$$\boldsymbol{t}_i \leftarrow \boldsymbol{t}_i \; + \boldsymbol{H}_i^{-1} \nabla_{\boldsymbol{t}_i} \mathcal{L}\,. \tag{4.8}$$

Here $\nabla_{\boldsymbol{w}_d} \mathcal{L}$ and $\nabla_{\boldsymbol{t}_i} \mathcal{L}$ are gradients of the log-likelihood (4.5) with respect to $\boldsymbol{w}_d$ and $\boldsymbol{t}_i$. The corresponding $\boldsymbol{H}$ matrices ($\boldsymbol{H}_d$ and $\boldsymbol{H}_i$) can be seen as approximations to conventional full

Hessian matrix in Newton-Raphson optimization (Bishop, 2006). These approximations are much smaller and faster to compute; and are proposed in (Povey et al., 2011a). See Appendix A for step-by-step derivation.

4.1.2 Limitations of the model

In a document collection, the most frequently occurring words are stop words which do not have the ability to semantically discriminate the documents. Moreover when using a large vocabulary of words (including the stop words), the number of parameters in the model increases and leads to over-fitting. To over come this, the model can be regularized.

A variant of SMM called subspace n-gram model (SnGM) was proposed for phonotactic language recognition (Soufifar et al., 2013), where the authors used ℓ_2 regularized model. This can be interpreted as obtaining maximum a posteriori (MAP) point estimates of the parameters with Gaussian prior. Further, it was observed in (Soufifar et al., 2013), that the embeddings ($\boldsymbol{w}_d \,\forall\, d$) exhibit Gaussian-like distribution across various dimensions, and the rows in $\boldsymbol{T}$ exhibit Laplace-like distribution (which does not comply with Gaussian prior assumption). Motivated by these observations and the desired property of sparsity in topic models, I propose to use ℓ_1 regularization for the rows in matrix $\boldsymbol{T}$, which can be seen as obtaining point MAP estimates with Laplace prior. ℓ_2 regularization is used for embeddings $\boldsymbol{w}_d \,\forall\, d$. The resulting model is called ℓ_1 SMM.

4.2 ℓ_1 SMM

By adding the respective ℓ_1 and ℓ_2 regularization terms to (4.5), the complete objective function becomes:

$$\mathcal{L} = \sum_{d=1}^{D} \underbrace{\Big[\sum_{i=1}^{V} x_{di} \log(\theta_{di}) - \frac{\lambda}{2} \|\boldsymbol{w}_d\|_2 \Big]}_{\mathcal{L}_d} - \omega \sum_{i=1}^{V} \|\boldsymbol{t}_i\|_1, \tag{4.9}$$

where ω and λ are the regularization weights for rows in $\boldsymbol{T}$ and $\boldsymbol{w}_d \,\forall\, d$ respectively; $\mathcal{L}_d$ denotes the log-likelihood per document with the regularization term for the document specific embedding. It is essential to regularize both $\boldsymbol{T}$ and $\boldsymbol{w}$. Otherwise, restricting the magnitude of one parameter will be compensated by dynamic range increase in the other during the iterative update steps (4.7) and (4.8).

Estimating the parameters of any ℓ_1 regularized function is not trivial, as it introduces discontinuities at points where the function crosses any axis. To address this, several optimization techniques were proposed (Andrew and Gao, 2007; Schmidt, 2010). One such technique, called as orthant-wise learning is explored in this work, as it could be translated in a straightforward way to the existing second order optimization scheme (4.8).

Orthant is a region in the n-dimensional space where the sign of the variables does not change. It is equivalent to quadrant in 2D and octant in 3D. An important property of any ℓ_1 regularized function is its differentiable nature over any given orthant. In general, for any ℓ_1 regularized convex objective function, if the initial point is in the same orthant as the minimum, then the simple Newton-Raphson updates will lead to the minimum[2]. In cases where the update steps need to cross the orthant to find the minimum, orthant-wise learning can be employed. Illustration of orthant-wise learning is presented in Appendix E.

4.2.1 Parameter estimation using orthant-wise learning

The gradient of the objective function in (4.9) with respect to $\boldsymbol{t}_i$ is given by:

$$\nabla_{\boldsymbol{t}_i}\mathcal{L} = \sum_{d=1}^{D}\Big(x_{di} - \theta_{di}\sum_{i=1}^{V} x_{di}\Big)\boldsymbol{w}_d^{\mathsf{T}} - \ \omega\ \text{sign}(\boldsymbol{t}_i)\,. \tag{4.10}$$

Here sign is the element-wise sign operation on the vector $\boldsymbol{t}_i$. At co-ordinates where the objective function is not differentiable (i.e., when any of the co-ordinates k in $\boldsymbol{t}_i$ equals to 0), its sub-gradient $\tilde{\nabla}\boldsymbol{t}_i$ is used:

$$\tilde{\nabla}_{t_{ik}}\mathcal{L} \triangleq \begin{cases} \nabla_{t_{ik}}\mathcal{L} + \omega, & t_{ik} = 0,\ \nabla_{t_{ik}}\mathcal{L} < -\omega \\ \nabla_{t_{ik}}\mathcal{L} - \omega, & t_{ik} = 0,\ \nabla_{t_{ik}}\mathcal{L} > \omega \\ 0, & t_{ik} = 0,\ |\nabla_{t_{ik}}\mathcal{L}| \leq \omega \\ \nabla_{t_{ik}}\mathcal{L}, & |t_{ik}| > 0\ . \end{cases} \tag{4.11}$$

Otherwise, $\tilde{\nabla}_{t_{ik}}\mathcal{L} = \nabla_{t_{ik}}\mathcal{L}$. The updates following Newton-Raphson like method require two things: (i) the search direction $\boldsymbol{d}$ that agrees with the direction of steepest ascent and, (ii) a step in the ascent direction that does not cross the point of non-differentiability. The search direction $\boldsymbol{d}_i$ is given by:

$$\boldsymbol{d}_i \triangleq \boldsymbol{H}_i^{-1}\tilde{\nabla}_{\boldsymbol{t}_i}\mathcal{L} \tag{4.12}$$

To ensure that the new estimates ($\boldsymbol{t}_i^{\text{new}}$) are along an ascent direction ($\boldsymbol{d}_i\nabla_{\boldsymbol{t}_i}\mathcal{L} > 0$), the co-ordinates in search direction $\boldsymbol{d}_i$ are set to zero, if the sign does not match with the co-ordinates in the steepest ascent $\nabla_{\boldsymbol{t}_i}\mathcal{L}$. This is called sign projection $\mathcal{P}_{\mathcal{S}}$, which is defined as:

$$\mathcal{P}_{\mathcal{S}}(\boldsymbol{d})_i \triangleq \begin{cases} d_{ik}, & \text{if} \quad d_{ik}(\tilde{\nabla}_{t_{ik}}\mathcal{L}) > 0\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{4.13}$$

Next, to ensure that the step does not cross the point of non-differentiability, the following orthant projection $\mathcal{P}_{\mathcal{O}}$ is applied:

$$\mathcal{P}_{\mathcal{O}}(\boldsymbol{t}+\boldsymbol{d})_i \triangleq \begin{cases} 0 & \text{if} \quad t_{ik}(t_{ik}+d_{ik}) < 0\,, \\ t_{ik}+d_{ik} & \text{otherwise}\,. \end{cases} \tag{4.14}$$

[2]In case of quadratic function, a single Newton-Raphson update will lead to minimum.

This orthant projection will set the co-ordinates in $\boldsymbol{t}_i^{\text{new}}$ to zero, if they differ in sign with $\boldsymbol{t}_i$. Finally, the update for $\boldsymbol{t}_i$ is given as follows:

$$\boldsymbol{t}_i \leftarrow \mathcal{P}_{\mathcal{O}}[\boldsymbol{t}_i + \mathcal{P}_{\mathcal{S}}[\,\boldsymbol{H}_i^{-1}\tilde{\nabla}_{\boldsymbol{t}_i}\mathcal{L}]\,]\,, \tag{4.15}$$

where $\boldsymbol{H}_i$ is computed as follows:

$$\boldsymbol{H}_i = -\left(\sum_{d=1}^{D} \max\left(x_{di}, \theta_{di} \sum_{i=1}^{V} x_{di}\right)\right) \boldsymbol{w}_d \boldsymbol{w}_d^{\mathsf{T}}\,. \tag{4.16}$$

Note that the $\boldsymbol{H}_i$ is not the exact second derivative[3] of $\tilde{\nabla}_{\boldsymbol{t}_i}\mathcal{L}$ but rather an intuitive approximation conceived by Povey (2009). This allows much faster convergence without the need for backtracking. The updates for every document embedding $\boldsymbol{w}_d$ are according to (4.7), with the following gradient:

$$\nabla_{\boldsymbol{w}_d}\mathcal{L} = \sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}}(x_{di} - \ \theta_{di} \sum_{i=1}^{V} x_{di}) - \lambda \boldsymbol{w}_d\,, \tag{4.17}$$

where $\boldsymbol{H}_d$ is computed as follows:

$$\boldsymbol{H}_d = -\sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}} \boldsymbol{t}_i \ \max\left(x_{di}, \theta_{di} \sum_{i=1}^{V} x_{di}\right) - \lambda \boldsymbol{I}\,. \tag{4.18}$$

If the updates of $\boldsymbol{T}$ or $\boldsymbol{w}$ fail to increase the objective function in (4.9), the update step is halved by backtracking[4]. Typically the model converges after 15 to 20 iterations. The complete training procedure for SMM is given in Algorithm 2.

Although the existing Newton-Raphson like optimization scheme helps the model converge in few iterations, it requires to make matrix inversions that has a time complexity of $\mathcal{O}(n^3)$. Further, the number of matrix inversion operations increase linearly with number of documents and also the size of vocabulary. The complexity in the existing optimization can be reduced by replacing the Newton-Raphson scheme with ADAGRAD (Duchi et al., 2011) or ADAM (Kingma and Ba, 2015). It was observed that ADAM converges to global optimum in convex optimization problems involving sparse data such as discrete word counts. The objective function of SMM is conditionally convex (or bi-convex); which means that given one set of parameters ($\boldsymbol{T}$), the objective is a convex function over other set of parameters ($\boldsymbol{w}_d$) and vice-versa. This makes ADAM optimization scheme suitable for SMM.

[3] The exact second derivative is given in (A.27) from Appendix A.

[4] In practice, the backtracking happens very rarely.

Algorithm 2: Training algorithm for SMM

initialize $\boldsymbol{m}$ to log uni-gram probabilities using (4.6)
initialize values in $\boldsymbol{T}$ from $\mathcal{N}(0, 0.001)$
initialize embeddings $\boldsymbol{w}_d \, \forall \, d$ to $\mathbf{0}$
repeat
  for $d = 1 \ldots D$ **do**
    compute $\mathcal{L}_d$ from (4.9)
    compute gradients of $\mathcal{L}_d$ w.r.t $\boldsymbol{w}_d$ using (4.17)
    compute $\boldsymbol{H}_d$ according to (4.18)
    update $\boldsymbol{w}_d$ using (4.7)
    while $\mathcal{L}_d$ *doesn't improve* **do**
      backtrack by halving the update step
    end
  end
  compute $\mathcal{L}$ using (4.9)
  for $i = 1 \ldots V$ **do**
    compute sub-gradients of $\mathcal{L}$ w.r.t $\boldsymbol{t}_i$ using (4.11)
    compute $\boldsymbol{H}_i$ according to (4.16)
    update $\boldsymbol{t}_i$ using orthant-wise learning according to (4.15)
  end
  while $\mathcal{L}$ *doesn't improve* **do**
    backtrack by halving the update step
  end
until *convergence or max_iterations*

4.3 ADAM optimization scheme for SMM

The Newton-Raphson like update steps from Eqs. (4.8) and Eqs. (4.7) are now replaced by the following:

$$\boldsymbol{w}_d \leftarrow \boldsymbol{w}_d + \eta \, (\frac{\hat{\boldsymbol{f}}_{wd}}{\sqrt{\hat{\boldsymbol{s}}_{wd}} + \epsilon}) \, \nabla_{\boldsymbol{w}_d} \mathcal{L} \, , \tag{4.19}$$

$$\boldsymbol{t}_i \leftarrow \mathcal{P}_{\mathcal{O}}(\boldsymbol{t}_i + \boldsymbol{d}_i) \, , \tag{4.20}$$

where,

$$\boldsymbol{d}_i = \eta \left(\frac{\hat{\boldsymbol{f}}_{td}}{\sqrt{\hat{\boldsymbol{s}}_{td}} + \epsilon} \right), \tag{4.21}$$

$$\mathcal{P}_{\mathcal{O}}(\boldsymbol{t} + \boldsymbol{d})_i \triangleq \begin{cases} 0 & \text{if} \quad t_{ik}(t_{ik} + d_{ik}) < 0, \\ t_{ik} + d_{ik} & \text{otherwise}\,. \end{cases} \tag{4.22}$$

Here η is learning rate, $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{s}}$ represent bias corrected first and second order moment estimate[5] of gradients (as required by ADAM), and $\mathcal{P}_{\mathcal{O}}$ represents orthant projection, assuring that the update step does not cross the point of non-differentiability. Unlike in (4.13), it is not required to apply the sign projection, because both gradient $\tilde{\nabla}_{\boldsymbol{t}_i}\mathcal{L}$ and step $\boldsymbol{d}_i$ point in the same direction (due to properties of ADAM). Later in § 4.4.2, we will compare the second order optimization (Newton-Raphson) with ADAM .

4.3.1 Extracting document embeddings

Once the model is trained, the embedding $\boldsymbol{w}_t$ for any (unseen) document $\boldsymbol{x}_t$ is extracted by keeping the model parameters $\{\boldsymbol{m}, \boldsymbol{T}\}$ fixed and using updates in (4.7) that maximize the regularized log-likelihood (objective) function. This procedure is identical to that of in training, except that the model parameters are not updated. In practice, the document embeddings are extracted[6] for both the training and test documents.

These embeddings are then used as input vectors for training classifier for the task of topic identification; or as an input for k-means clustering algorithm.

4.4 Experiments and results

4.4.1 Dataset

The experiments were conducted on the *20 Newsgroups*[7] dataset as it is well-studied with several benchmark baseline systems. I have used the preprocessed version (`20news-bydate-matlab`) as used in (Zhu and Xing, 2011; Lacoste-Julien et al., 2008). It contains 18775 documents in 20 categories. The training set consists of 11269 documents with a vocabulary of 53975 words and the test set consists of 7505 documents.

4.4.2 Comparison of Newton-Raphson with ADAM optimization

Fig. 4.3 shows the improvement of log-likelihood of the *20Newsgroups* training data over the iterations with both the optimization schemes. We can see that ADAM takes many more

[5] See appendix D

[6] It can be interpreted as feature extraction.

[7] http://qwone.com/~jason/20Newsgroups/

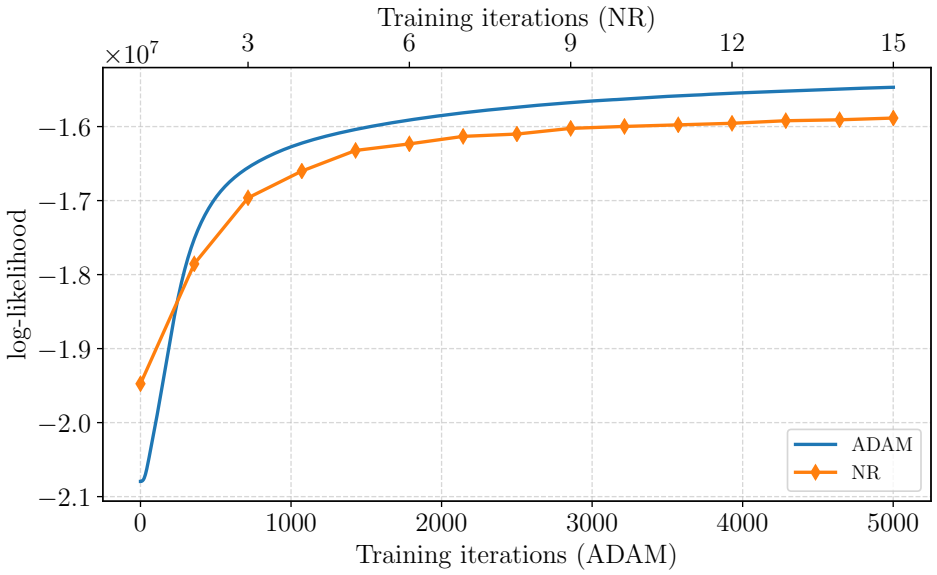


Figure 4.3: Convergence of ℓ_1 SMM with Newton-Raphson (NR) and ADAM optimization schemes. Model was trained on *20Newsgroups* data with $K = 100, \lambda = 1e-4, \omega = 1e-1$.

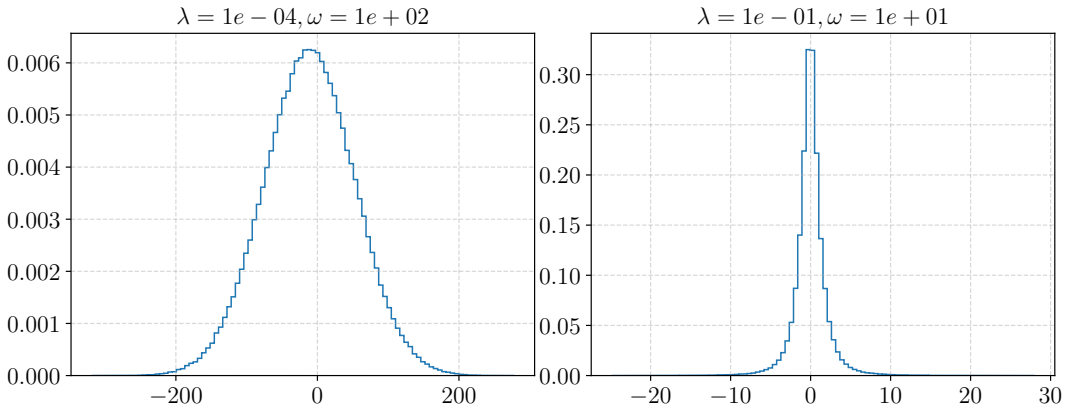


Figure 4.4: Histogram of embeddings extracted from ℓ_1 SMM.

iterations to converge as compared to Newton-Raphson, but it is computationally cheaper as it does not involve any matrix inversions. On a single CPU, ADAM is approximately 10 times slower than Newton-Raphson, but consumes one-third memory. The advantage of ADAM can be seen on a GPU while performing batch-wise stochastic training on a large dataset. Since ADAM consumes less memory, large batches can be loaded onto GPU which helps in faster training.

4.4.3 Analysis of model parameters

This section presents the analysis of model parameters of ℓ_1 and ℓ_2 SMM. All the experiments were done using Newton-Raphson optimization scheme with the hyper-parameters: $\lambda = 1e{-}04$, and embedding dimension $K = 100$.

The Fig. 4.4, shows that the document embeddings exhibit Gaussian like distribution. This influenced the choice of classifier for document classification task (§ 4.4.4.2) and clustering algorithm for clustering task (§ 4.4.5).

Fig. 4.5 shows various histograms of values from $\boldsymbol{T}$ for various regularization weights ω. The distribution of values from $\boldsymbol{T}$ matrix in Fig. 4.5 suggests that Laplace prior was an appropriate choice. The ℓ_1 regularization with orthant-wise learning enforces sparsity in the matrix ($\boldsymbol{T}$), and we can see that sparsity is directly proportional to the ℓ_1 regularization weight. In case of ℓ_2 SMM, even higher values of ω does not introduce any sparsity, however all the parameters and densely packed around zero.

4.4.4 Document classification task

In this task, the document representations are evaluated by using them as input features for a classifier during training and testing phases. I have used only linear classifiers, as they are faster to train and do not learn any additional non-linear transformations of the input representations.

4.4.4.1 Baseline systems for classification

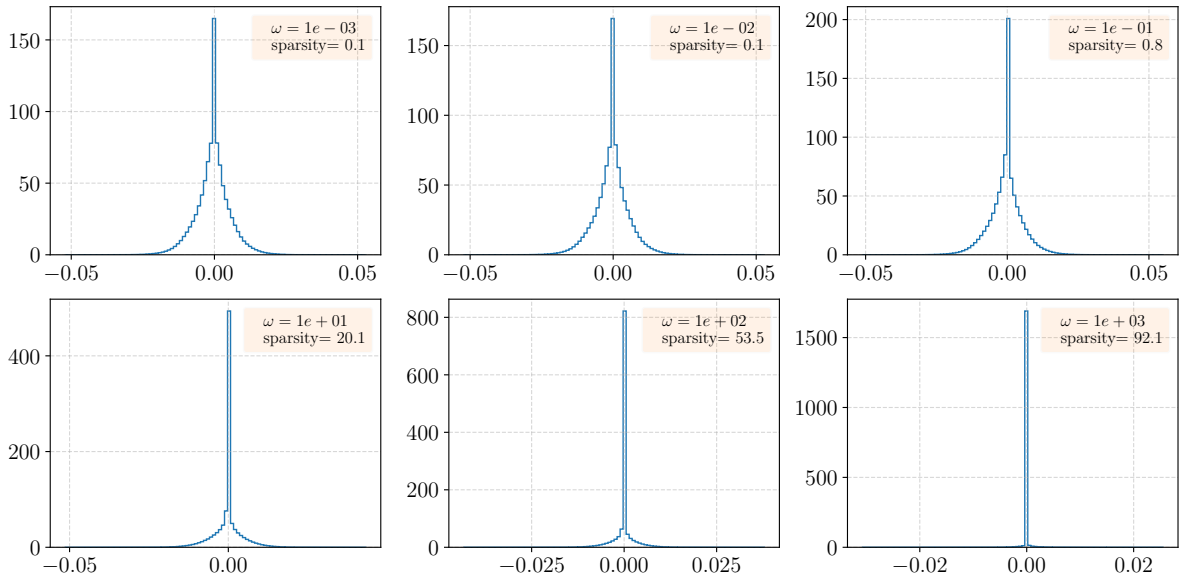
I have used two baseline systems: latent Dirichlet allocation (LDA) (Blei et al., 2003), and sparse topical coding (STC) (Zhu and Xing, 2011), which are unsupervised topic models. LDA was chosen because it is the most popular and well understood topic model. STC is a superior to LDA and also has sparse model parameters, hence it acts a baseline comparison with the proposed sparse ℓ_1 SMM. They are trained only on documents from the training set; then the document representations are extracted for both training and test sets. Following earlier research (Zhu and Xing, 2011), I have chosen support vector machines (SVM) as the choice of classifier for these representations. LDA[8], and STC[9] were trained using publicly available source code.

4.4.4.2 Proposed systems for classification

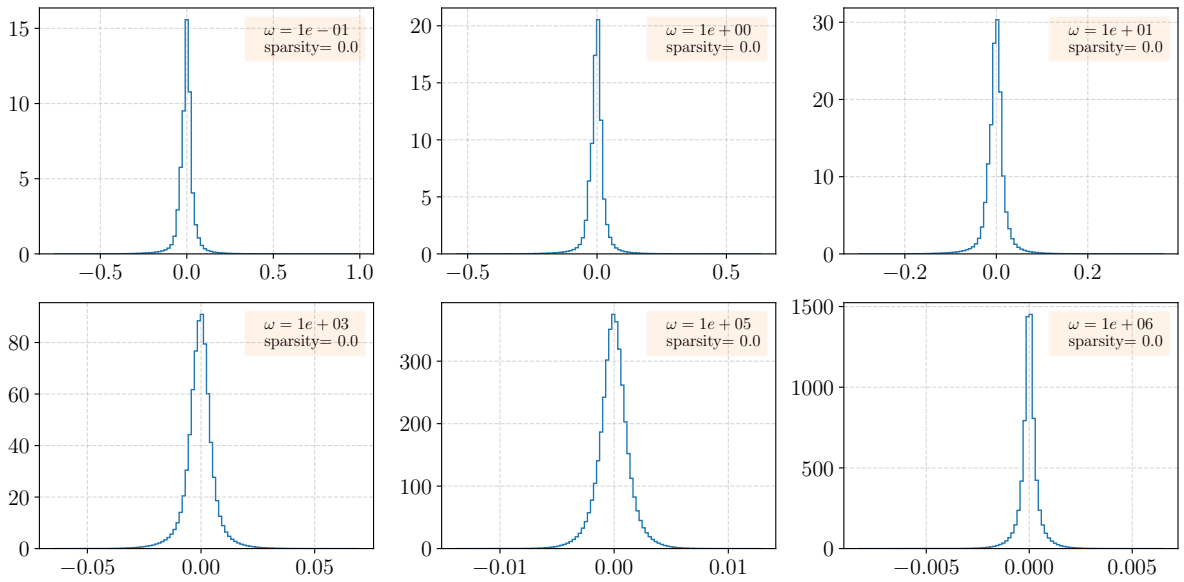
The proposed systems for classification use document embeddings extracted from SMM. I used two variants of SMM; the first one is the proposed ℓ_1 SMM and the second one is ℓ_2 SMM

[8] https://github.com/blei-lab

[9] http://ml.cs.tsinghua.edu.cn/~jun



(a) ℓ_1 regularization.



(b) ℓ_2 regularization.

Figure 4.5: Histograms showing the distribution of values from the matrix $\boldsymbol{T}$ for various regularization weights ω. The other hyper-parameters, embedding dimension $K = 100$ and $\lambda = 1e - 04$.

i.e., ℓ_2 regularization over bases matrix $\boldsymbol{T}$. Since the document embeddings exhibit Gaussian-like distribution (Fig. 4.5), I have used simple generative Gaussian linear classifier (GLC),

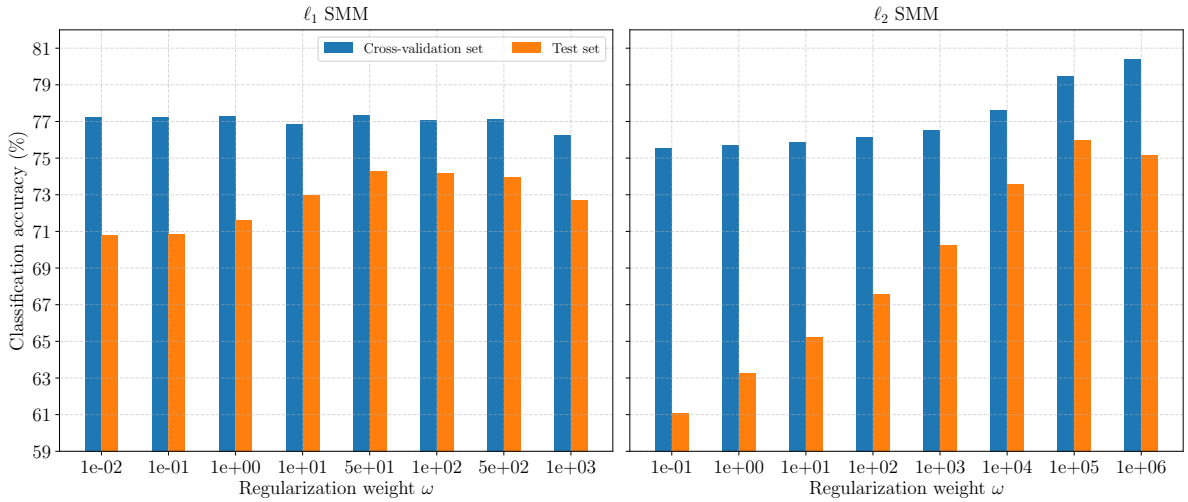


Figure 4.6: Classification accuracy on *20Newsgroups* data for ℓ_1 and ℓ_2 SMM with various regularization weights ω. The other hyper-parameters, embedding dimension $K = 100$ and $\lambda = 1e-04$.

where every class is Gaussian distributed with a specific mean and shares a common covariance matrix (Bishop, 2006).

First, I give the comparison between ℓ_1 and ℓ_2 SMM based classification systems. Fig. 4.6 shows the classification accuracy on the test set, for various values of ω (regularization coefficient of $\boldsymbol{T}$) with embedding dimension $K = 100$. For the purpose of illustration, the regularization coefficient of embeddings λ is fixed to 10^{-4}. We can observe two things from the Fig. 4.6

1. The absolute difference in accuracy between cross-validation and test sets for ℓ_1 SMM is lower as compared to ℓ_2 SMM. This suggests that ℓ_1 SMM generalizes better as compared to ℓ_2 SMM.

2. As we increase the ℓ_2 regularization weight (ω) the performance of ℓ_2 SMM increases. Note that higher regularization forces the parameters to be close to zero (Fig. 4.5), which suggests the suitability of ℓ_1 regularization as it explicitly introduces zeros.

Next, the proposed systems are compared against rest of the baseline systems. The corresponding classification scores are presented in Table 4.1 along with the latent variable dimension K. Additionally, the results are also compared with Discriminative LDA (DiscLDA) (Lacoste-Julien et al., 2008) and max margin supervised STC (MedSTC) (Zhu and Xing, 2011). Detailed comparison of STC and its variants along with various other models is given in (Zhu and Xing, 2011). All the classification results presented are the best values that are obtained by tuning on the development (cross-validation) set. It is important to note that DiscLDA and MedSTC which achieve better classification results are supervised models i.e., topic label information is

Table 4.1: Comparison of classification accuracy (in %) across various systems based on supervised and unsupervised topic models.

Model	**Classifier**	**Embedding dimension** K	**Accuracy** (%)
LDA	SVM	110	75.0
STC	SVM	90	74.0
ℓ_2 SMM	GLC	100	75.1
ℓ_2 SMM	GLC	200	**77.1**
ℓ_2 SMM	GLC	300	67.5
ℓ_1 SMM	GLC	100	74.3
ℓ_1 SMM	GLC	200	**76.1**
ℓ_1 SMM	GLC	300	**76.7**
DiscLDA (Lacoste-Julien et al., 2008)	SVM	100	80.0
MedSTC (Zhu and Xing, 2011)	SVM	100	81.0

incorporated while obtaining the latent vector representation; whereas their counterparts, LDA and STC are completely unsupervised models like SMM. From these results in Table 4.1, it can be observed that document representations (embeddings) obtained from ℓ_1 SMM together with simple generative linear classifier are superior when compared to the baseline systems that use discriminative classifier. Note that, although ℓ_2 SMM is slightly better than ℓ_1 SMM, it tends to over-fit as the embedding dimension is increases. The over-fitting nature of ℓ_2 SMM can also be seen in Fig. 4.6. Further, the classification accuracy of ℓ_1 SMM based system increases with the increase in dimensionality of the latent variable (embedding); which can be seen from Table. 4.1. However, this trend was seen neither for STC nor LDA (Zhu and Xing, 2011).

4.4.5 Document clustering task

In this task, the document embeddings are clustered and then evaluated using normalized mutual information (NMI). Since the *20 Newsgroups* dataset has 20 topics, the number of clusters are set to 20. This allows a simpler interpretation of NMI scores, i.e., "how much information about the true classes is inferred from the given clusters?".

Since, the document embeddings exhibit Gaussian-like distribution, I used k-means clustering algorithm with a fixed set of 20 clusters. Document embeddings extracted from entire dataset (training + test sets) were clustered, while keeping the model parameters $\{\boldsymbol{m}, \boldsymbol{T}\}$ trained only on the training set. This is to maintain consistency with the classification experi-

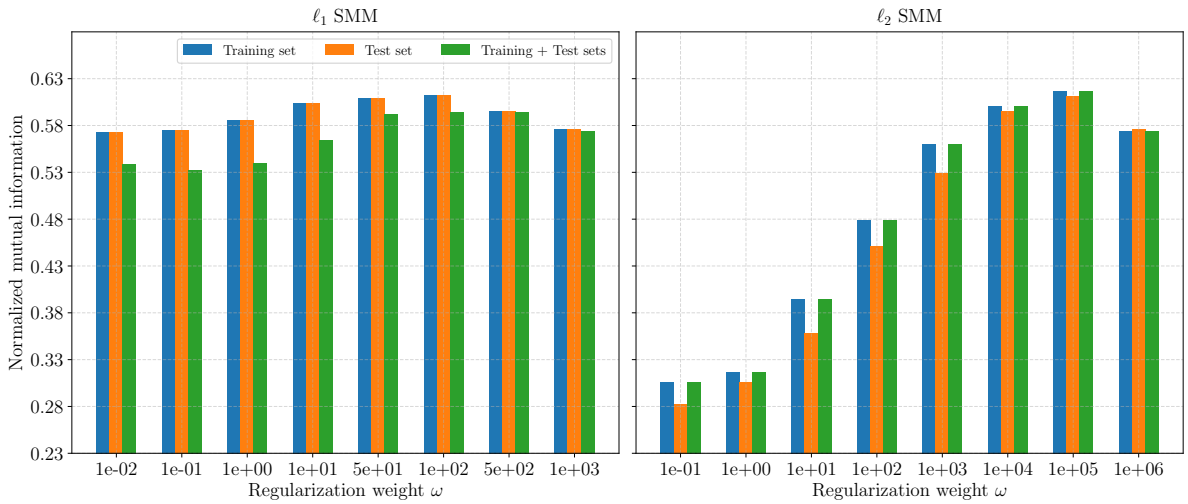


Figure 4.7: Normalized mutual information between the clusters and true classes of *20Newsgroups* data for ℓ_1 and ℓ_2 SMM with various regularization weights ω. The other hyperparameters are $K = 100$ and $\lambda = 1e - 04$.

Table 4.2: Comparison of average NMI scores of other systems with ℓ_1 SMM $\omega = 1e + 02$ and ℓ_2 SMM with $\omega = 1e + 06$. $\lambda = 1e - 04$, embedding dimension $K = 100$.

Model	ℓ_2 SMM	ℓ_1 SMM	LDA+k-means	LDA (naive)	STC+k-means
NMI	**0.61**	**0.58**	0.38	0.57	0.35

ments. The clustering was performed with 5 random initializations of k-means and the average of NMI scores are reported.

First, the NMI scores of ℓ_1 and ℓ_2 SMM based clustering systems are compared. Table 4.7 gives these scores along with various regularization weights ω. It can be concluded from these NMI scores that, the clusters obtained using document embeddings from ℓ_1 SMM are consistently stable as compared the ones from ℓ_2 SMM. This trend is observed across various values of ω, reinforcing the suitability of Laplace prior over the rows in matrix $\boldsymbol{T}$.

In Table 4.2, the comparison of NMI scores from proposed SMM based k-means clustering with other techniques are presented. In LDA (naive), the model is trained with 20 latent topics, i.e., the document representations are of 20 dimensions and live in 19 dimensional simplex. The cluster assignment was based on the largest value in each document vector. It can be seen that the proposed SMM performs better at clustering and classification at the same time with the same model (i.e., with the exact same model parameters including embedding dimension).

4.4.6 Topic discovery using SMM

In an unsupervised scenario, it is possible to obtain a set of words for each cluster that represent or discriminate it from other clusters. These set of words can describe the hidden (latent) topics in corpus. SMM can be used for topic discovery, i.e., the document embeddings extracted from SMM together with clusters obtained from k-means can bring out the hidden topics. One simple way is to subtract the global mean from the cluster mean of embeddings ($\boldsymbol{w}_d$) and project the resulting vector on to the bases matrix $\boldsymbol{T}$ and find the indices of large positive values. These indices corresponds to the words for which the probabilities significantly increase as compared to the average distribution over words for the given cluster. Table 4.3 shows an example of words from all the 20 clusters obtained using k-means for ℓ_1 SMM with $\lambda = 1e-04, \omega = 1e+01$ and embedding dimension $K = 100$.

The true classes in *20Newsgroups* corpus is given in the Table 4.4. We can observe that the clusters obtained using k-means with ℓ_1 SMM corresponds to most of the true topics / categories.

4.4.7 Discussion

SMM is an unsupervised model trained iteratively by optimizing the log-likelihood of the data; it does not necessarily correlate with the performance of topic ID. It is valid for LDA, STC 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, because a fully converged model may not yield embeddings optimal for downstream classification task.

In case of large-scale pre-trained models such as ULMFiT (Howard and Ruder, 2018) or BERT (Devlin et al., 2019), it is required to fine-tune the model for a particular dataset and classification task.

Fig. 4.8 illustrates the importance of early stopping mechanism. The grey line indicate the log-likelihood of the data, which we aim to maximize during training. The blue and orange curves represent the classification accuracy on cross-validation and test sets obtained at regular checkpoints during the training. Note that as the model is close to convergence, the embeddings results in poor classification performance. The ESM is computationally expensive because one needs to extract embeddings and train a classifier several times (for every checkpoint) during training. In an ideal scenario, one can let the model converge and use the embeddings directly for classification without any ESM or fine-tuning. In the next chapters, we will show that this is possible with the help of Bayesian modelling.

The document embeddings estimated from SMM are only point estimates. Given the bases matrix, each embedding captures the uni-gram probability distribution of words in the docu-

Table 4.3: Top 5 significant words representing 20 clusters.

acceleration	preferably	scotia	autoexec	xlib
suspension	architecture	sluggo	windows	widget
wagon	databases	compuserv	exe	parameter
tires	publisher	nursery	icons	openwindows
chevy	blvd	pruden	ini	xview
sale	waco	sacred	physicians	income
packaging	atf	worship	patients	socialism
obo	fbi	christianity	therapy	abortion
shipping	convicted	atheist	infection	welfare
cod	koresh	prophet	diagnosed	cramer
murders	privacy	resistor	hockey	nubus
criminals	encryption	amplifier	potvin	quadra
firearm	denning	resistors	leafs	meg
handguns	clipper	volt	nhl	slots
criminal	crypto	voltage	playoff	adapter
rbi	israeli	compute	spacecraft	zx
dodgers	lebanon	algorithms	lunar	bikes
hitters	occupied	polygon	moon	motorcycle
pitcher	palestinians	shareware	exploration	riding
pitching	palestinian	surfaces	orbit	bike

Table 4.4: Topics in *20Newsgroups* dataset

comp.graphics	rec.autos	sci.crypt
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med
comp.sys.mac.hardware	rec.sport.hockey	sci.space
comp.windows.x		
misc.forsale	talk.politics.misc	talk.religion.misc
	talk.politics.guns	alt.atheism
	talk.politics.mideast	soc.religion.christian

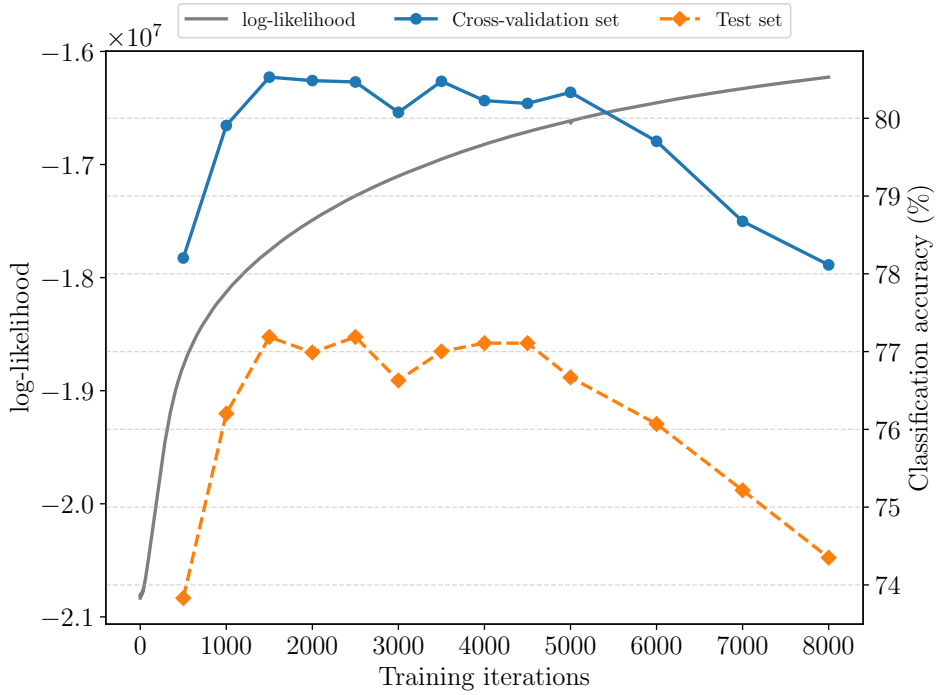


Figure 4.8: Illustrating the importance of early stopping.

ment. The embedding extraction is a iterative process, and after each iteration, the embedding is estimated to better explain the observed data (maximizes the log-likelihood). For longer documents, the extracted embeddings are robust (the embedding posterior distribution is peaky) as the uni-gram probabilities can be estimated from large number of observed words. The problem arises when embeddings are estimated on shorter and ambiguous documents. There exists many solutions (embeddings) that can explain the observed data. The embedding posterior distribution in this case is relatively flat, it means there exists many embeddings that can explain the same document. When training a classifier, the uncertain embeddings are outliers. For example, in case of generative Gaussian linear classifier, the uncertain embeddings will move the class mean away from the true mean. The uncertainty in embeddings can be estimate with the help of Bayesian learning. This is discussed in detail in Chapter 5, where Bayesian SMM is proposed for learning document embeddings along with their uncertainties, that are represented using Gaussian distribution. Chapter 6 shows how these uncertainties can be exploited in a classifier for the task of topic identification.

4.5 Summary and conclusions

This chapter presented the application of SMM to learn document embeddings (representations). Further, a novel variant of SMM was proposed, that based on ℓ_1 regularization of its

model parameters. The resulting ℓ_1 regularized objective function is optimized with the help of orthant-wise learning; which also introduced sparsity into the model parameters. With the help of linear classifiers, the obtained document representations from ℓ_1 SMM achieved better results in topic identification task when compared to popular topic models such as LDA and STC. A faster optimization scheme based on ADAM was proposed and also its adaptation to orthant-wise learning.

The analysis on the uncertainty of embeddings motivated the need for Bayesian modelling, which will be discussed in the following chapter.

Chapter 5

Learning document representations along with their uncertainties

This chapter presents a new model called Bayesian subspace multinomial model (Bayesian SMM). This model aims to overcome the major limitation of subspace multinomial model, that we have seen in the last chapter - by "representing document embeddings in the form of Gaussian distributions, thereby capturing the uncertainty in the estimates". The benefits of modelling uncertainty is reflected in the analysis and results. Additionally, this chapter will also address the problem of intractability that appears while performing variational inference in mixed-logit models (Bishop, 2006; Depraetere and Vandebroek, 2017).

5.1 Bayesian subspace multinomial model

Bayesian SMM is a generative model for the bag-of-words representation of documents, and the corresponding graphical model is depicted in Fig. 5.1. This generative probabilistic model assumes that the training data (i.e,. the vector of word counts $\boldsymbol{x}_d$) were generated as follows:

For each document $d = 1 \dots D$, a K-dimensional latent vector (document embedding) $\boldsymbol{w}_d$ is generated from Gaussian prior with mean $\boldsymbol{\mu} = 0$ and precision λ:

$$p_0 = p(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \,|\, \mathbf{0}\,, (\lambda \boldsymbol{I})^{-1}), \tag{5.1}$$

The latent vector $\boldsymbol{w}_d$ is a low dimensional representation ($K \ll V$) 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 $\boldsymbol{\theta}_d \in \triangle^{\mathsf{V}-1}$ is calculated as:

$$\boldsymbol{\eta}_d = \boldsymbol{m} + \boldsymbol{T}\,\boldsymbol{w}_d \tag{5.2}$$

$$\boldsymbol{\theta}_d = \text{softmax}(\boldsymbol{\eta}_d), \tag{5.3}$$

where $\boldsymbol{m} \in \mathbb{R}^{V \times 1}$ and $\boldsymbol{T} \in \mathbb{R}^{V \times K}$ are the parameters of the model. The vector $\boldsymbol{m}$ known as universal background model represents (or bias) log uni-gram probabilities of words. $\boldsymbol{T}$ known

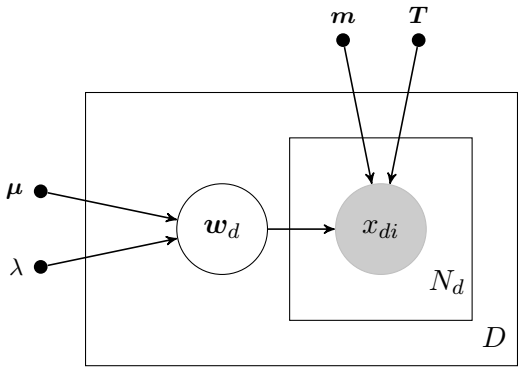


Figure 5.1: Graphical representation for Bayesian subspace multinomial model, where arrows show the dependency between the variables. The shaded circle $\boldsymbol{x}_d$ represents the observed document (word counts) and $\boldsymbol{w}_d$ represents the document-specific latent variable.

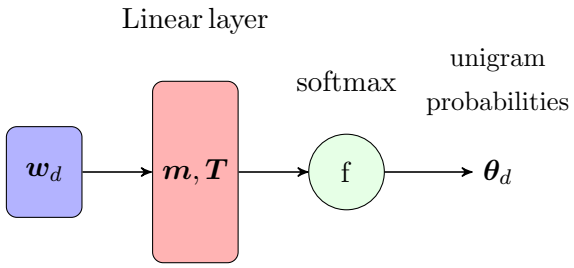


Figure 5.2: Alternative representation of Bayesian SMM, where $\boldsymbol{m}, \boldsymbol{T}$ represent the bias and weights. $q(\boldsymbol{w}_d)$ is the posterior distribution of the document-specific latent variable and $\boldsymbol{x}_d$ is the observed document (word counts).

as total variability (or weight) matrix (Kockmann et al., 2010; Dehak et al., 2011) is a low-rank matrix defining subspace of document specific distributions.

Finally, for each document, a vector of word counts $\boldsymbol{x}_d$ is sampled from multinomial distribution:

$$\boldsymbol{x}_d \sim \text{Multi}(\boldsymbol{\theta}_d, N_d), \tag{5.4}$$

where N_d is the number of words in document d.

$\boldsymbol{\eta}$ from (5.3) represents the *natural parameters* of the Multinomial distribution. Further, we can see that our model is linear in the space of natural parameters (5.2). Note that the parameters of any probability distribution under the *exponential family* can be expressed in terms of its *natural parameters* (Bishop, 2006).

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

The posterior distribution for a document embedding $\boldsymbol{w}_d$ is obtained by using Bayes' rule. For clarity, explicit conditioning on $\boldsymbol{T}$ and $\boldsymbol{m}$ is omitted in the subsequent equations.

$$p(\boldsymbol{w}_d|\boldsymbol{x}_d) = \frac{p(\boldsymbol{x}_d|\boldsymbol{w}_d)p(\boldsymbol{w}_d)}{\int p(\boldsymbol{x}_d|\boldsymbol{w}_d)p(\boldsymbol{w}_d)\,\mathrm{d}\boldsymbol{w}_d}. \tag{5.5}$$

In numerator of (5.5), $p(\boldsymbol{w}_d)$ represents prior distribution of document embeddings, which is given by (5.1) and $p(\boldsymbol{x}_d|\boldsymbol{w}_d)$ represents the likelihood of observed data. According to the generative process, every document $\boldsymbol{x}_d$ is assumed to be a sample from multinomial distribution (5.4), hence the log-likelihood is computed as:

$$\ln p(\boldsymbol{x}_d|\boldsymbol{w}_d) = \sum_{i=1}^{V} x_{di} \ln \theta_{di}, \tag{5.6}$$

$$= \sum_{i=1}^{V} x_{di} \log \left(\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}} \right), \tag{5.7}$$

$$= \sum_{i=1}^{V} x_{di} \left[(m_i + \boldsymbol{t}_i \boldsymbol{w}_d) - \log \left(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} \right) \right], \tag{5.8}$$

where $\boldsymbol{t}_i$ represents a row in matrix $\boldsymbol{T}$. The problem arises while computing the denominator in (5.5). It involves solving integral over product of likelihood containing softmax function and Gaussian distribution:

$$\int p(\boldsymbol{x}_d|\boldsymbol{w}_d) p(\boldsymbol{w}_d) \, \mathrm{d}\boldsymbol{w}_d = \int \left(\sum_{i=1}^{V} \left[\frac{\exp\{m_i + \boldsymbol{t}_i \, \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}} \right]^{x_{di}} \right) \left(\mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{0}, \mathrm{diag}(\lambda)^{-1}) \right) \mathrm{d}\boldsymbol{w}_d \tag{5.9}$$

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 (Bishop, 2006). 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 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 (see § 3.1.1) to minimize the KL divergence. We express the log marginal (evidence) of the data as:

$$\ln p(\boldsymbol{x}_d) = \mathbb{E}_q[\ln p(\boldsymbol{x}_d, \boldsymbol{w}_d)] + \mathrm{H}[q] + D_{\mathrm{KL}}(q \,||\, p), \tag{5.10}$$

$$= \mathcal{L}(q_d) + D_{\mathrm{KL}}(q \,||\, p). \tag{5.11}$$

Here $\mathrm{H}[q]$ represents the entropy of $q(\boldsymbol{w}_d)$. Given the data $\boldsymbol{x}_d$ and model parameters, $\ln p(\boldsymbol{x}_d)$ is a constant, and $D_{\mathrm{KL}}(q \,||\, p)$ can be minimized by maximizing $\mathcal{L}(q_d)$, which is known as *Evidence Lower BOund* (ELBO) for a document. See Chapter 3.1.1 for the derivation of VB formulation in general.

5.2 Variational Bayes

Using the VB framework, this section explains and derives the procedure for estimating model parameters $\{\boldsymbol{m}, \boldsymbol{T}\}$ and inferring the variational distribution, $q(\boldsymbol{w}_d)$. Before proceeding, note that the model assumes that all documents and the corresponding document embeddings (latent variables) are independent. This can be seen from the graphical model in Fig. 5.1. Hence, the inference is derived only for one document embedding $\boldsymbol{w}$, given observed vector of word counts $\boldsymbol{x}$. For brevity, the document suffix d is omitted in further.

The variational distribution $q(\boldsymbol{w})$ is chosen to be Gaussian, with mean $\boldsymbol{\nu}$ and precision $\boldsymbol{\Gamma}$, i.e., $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\nu}, \boldsymbol{\Gamma}^{-1})$. An analytical form of $\mathcal{L}(q)$ is required for the optimization. We proceed as follows:

$$\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}_q[\ln p(\boldsymbol{x}, \boldsymbol{w})] \; + \; \mathrm{H}[q], && (5.12)\\
&= \mathbb{E}_q[\ln p(\boldsymbol{x} \mid \boldsymbol{w})] + \mathbb{E}_q[\ln p(\boldsymbol{w})] + H[q], && (5.13)\\
&= \underbrace{\mathbb{E}_q[\ln p(\boldsymbol{x} \mid \boldsymbol{w})]}_{\mathsf{A}} - \underbrace{D_{\mathrm{KL}}(q \,||\, p_0)}_{\mathsf{B}} && (5.14)
\end{aligned}$$

The term B in (5.14) is the KL divergence from the variational distribution $q(\boldsymbol{w})$ to the document-independent prior (5.1), which can be computed analytically (Petersen and Pedersen, 2012) as:

$$D_{\mathrm{KL}}(q \,||\, p_0) = \frac{1}{2}\Big[\lambda \,\mathrm{tr}\Big(\boldsymbol{\Gamma}^{-1}\Big) + \ln|\boldsymbol{\Gamma}| - K \ln \lambda + \lambda \boldsymbol{\nu}^{\mathsf{T}} \boldsymbol{\nu} - K\Big], \tag{5.15}$$

where K denotes the dimension of document embedding. See Appendix B.1 for step-by-step derivation. The term B from (5.14) is the expectation over log-likelihood of a document (5.8):

$$\mathbb{E}_q[\ln p(\boldsymbol{x} \mid \boldsymbol{w})] = \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}) - \underbrace{\mathbb{E}_q\Big[\ln \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}\} \right) \Big]}_{\mathcal{F}} \right] \tag{5.16}$$

where

$$\mathcal{F} = \mathbb{E}_q[\, \ln \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}\} \right)] \tag{5.17}$$

$\mathcal{F}$ (5.17) involves solving the expectation over log-sum-exp function which is intractable. This kind of expectation appears when dealing with variational inference in mixed-logit models such as logistic regression (Blei and Lafferty, 2005; Depraetere and Vandebroek, 2017). We will present two ways of addressing this intractability. The first approach uses Jensen's inequality to form an upper bound on $\mathcal{F}$, whereas the second approach approximates $\mathcal{F}$ using Monte-Carlo samples via re-parametrization.

5.2.1 Jensen's inequality

Since logarithm is a concave function, we can use Jensen's inequality on (5.17) and obtain the following:

$$\mathbb{E}_q[\ln\left(\sum_{j=1}^{V}\exp\{m_j+\boldsymbol{t}_j\boldsymbol{w}\}\right)] \leq \ln\left(\mathbb{E}_q[\sum_{j=1}^{V}\exp\{m_j+\boldsymbol{t}_j\boldsymbol{w}\}]\right) \tag{5.18}$$

$$= \ln\left(\sum_{j=1}^{V}\exp\{m_j\}\,\mathbb{E}_q[\exp(\boldsymbol{t}_j\boldsymbol{w})]\right) \tag{5.19}$$

$$= \ln\left(\sum_{j=1}^{V}\exp\{m_j+\boldsymbol{t}_j\boldsymbol{\nu}+\frac{1}{2}\boldsymbol{t}_j\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_j^{\mathsf{T}}\}\right) \tag{5.20}$$

Note that (5.20) forms an upper bound on $\mathcal{F}$. Now combining (5.20), (5.16), and (5.14), we have a lower-bound on $\mathcal{L}(q)$:

$$\mathcal{L}(q_d) \geq -D_{\mathrm{KL}}(q_d \,||\, p_0) + \sum_{i=1}^{V} x_i\left[(m_i+\boldsymbol{t}_i\boldsymbol{\nu}) - \ln\left(\sum_{j=1}^{V}\exp\{m_j+\boldsymbol{t}_j\boldsymbol{\nu}+\frac{1}{2}\boldsymbol{t}_j\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_j^{\mathsf{T}}\}\right)\right]. \tag{5.21}$$

See appendix B.1.1 for the step-by-step derivation of (5.21).

5.2.2 Approximation using Monte-Carlo samples via re-parametrization trick

We can approximate $\mathcal{F}$ (5.17) 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 (Paisley et al., 2012). To overcome this, we will re-parametrize the random variable $\boldsymbol{w}$ by introducing a differentiable function g over another random variable $\boldsymbol{\epsilon}$ (Kingma and Welling, 2014). If $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, then:

$$\boldsymbol{w} = g(\boldsymbol{\epsilon}) = \boldsymbol{\nu} + \boldsymbol{L}\,\boldsymbol{\epsilon}, \tag{5.22}$$

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

$$\mathcal{F} \approx \frac{1}{R}\sum_{r=1}^{R}\ln\left(\sum_{j=1}^{V}\exp\{m_j+\boldsymbol{t}_j\,g(\tilde{\boldsymbol{\epsilon}}_r)\}\right), \tag{5.23}$$

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

Blei and Lafferty (2005) encountered the same problem while performing variational inference for correlated topic model (CTM), and used first-order Taylor series approximation for

[1] See Appendix B.1.2 for more details.

$\mathcal{F}$. This in-turn became a lower bound on $\mathcal{L}(q)$. The problem with first-order Taylor series approximation is illustrated in Chapter 3.2.3.1.

The re-parametrization trick for approximating $\mathcal{F}$ was also used in neural variational document model (Miao et al., 2016). There are similar approximation techniques based on Quasi Monte Carlo sampling (Depraetere and Vandebroek, 2017).

Combining (5.15), (5.16) and (5.23), we get the approximation to $\mathcal{L}(q)$. We will introduce back the document suffix d, to make the notation explicit:

$$\mathcal{L}_{\mathsf{RP}}(q_d) \approx -D_{\mathrm{KL}}(q_d \mid\mid p_0) + \sum_{i=1}^{V} x_{di} \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}_d) - \frac{1}{R} \sum_{r=1}^{R} \ln \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \, g(\, \tilde{\boldsymbol{\epsilon}}_{dr}\,)\} \right) \right]. \quad (5.24)$$

For the entire data $\boldsymbol{X}$, the complete ELBO will be simply the summation over all the documents, i.e., $\sum_d \mathcal{L}(q_d)$. Now that the analytical form approximating ELBO is obtained, we will explain the training procedure in the next section.

The following sections continue the discussion with the objective function (5.24) derived using the re-parametrization trick. However, in the § 5.5.4 we provide the empirical comparison with the model that uses Jensen's inequality.

5.3 Training

The variational Bayes (VB) training procedure for Bayesian SMM is stochastic because of the sampling involved in the re-parametrization trick (5.22). Like the standard VB approach (Bishop, 2006), 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:

$$\mathcal{L}_{\mathsf{RP}} = \sum_{d=1}^{D} \mathcal{L}_{\mathsf{RP}}(q_d) - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1 \,, \quad (5.25)$$

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 (Kesiraju et al., 2016). This can also be seen as obtaining a maximum a posteriori estimate of $\boldsymbol{T}$ with Laplace priors.

5.3.1 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 $\mathcal{N}(\mathbf{0}, \mathrm{diag}(0.1))$. Later in § 5.5.2, we will show that initializing the posterior to a sharper Gaussian distribution helps to speed up the convergence.

5.3.2 Optimization

The gradient-based updates are done by ADAM optimization scheme (Kingma and Ba, 2015); in addition to the following tricks. We simplified the variational distribution $q(\boldsymbol{w})$ by making its precision matrix $\boldsymbol{\Gamma}$ diagonal. Note that this is not a theoretical limitation but only a simplification. Further, while updating it, we used log standard deviation parametrization, which ensures that the variance is always positive:

$$\boldsymbol{\Gamma}^{-1} = \text{diag}(\exp\{2\,\boldsymbol{\varsigma}\}). \tag{5.26}$$

The gradients of the objective (5.24) w.r.t. the mean $\boldsymbol{\nu}$ is given as follows:

$$\nabla_{\boldsymbol{\nu}} \mathcal{L}_{\text{RP}}(q_d) = \Big[\sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}}(x_i - \frac{1}{R}\sum_{r=1}^{R}\theta_{ir}\sum_{k=1}^{V} x_k)\Big] - \lambda\boldsymbol{\nu} \tag{5.27}$$

where

$$\theta_{ir} = \frac{\exp\{m_i + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}} \tag{5.28}$$

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

$$\nabla_{\boldsymbol{\varsigma}} \mathcal{L}_{\text{RP}} = \mathbf{1} - \lambda\exp\{2\boldsymbol{\varsigma}\} - \sum_{k=1}^{V} x_k \frac{1}{R}\sum_{r=1}^{R}\sum_{i=1}^{V}\theta_{ir}\boldsymbol{t}_i^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}\} \odot \boldsymbol{\epsilon}_r, \tag{5.29}$$

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 (5.25) discontinuous i.e., non-differentiable at points where it crosses the orthant. Hence, we used sub-gradients and employed orthant-wise learning (Andrew and Gao, 2007). The gradient of the objective (5.25) w.r.t. a row $\boldsymbol{t}_i$ in matrix $\boldsymbol{T}$ is computed as follows:

$$\nabla_{\boldsymbol{t}_i} \mathcal{L}_{\text{RP}} = -\omega\,\text{sign}(\boldsymbol{t}_i) + \sum_{d=1}^{D}\Bigg[x_{di}\boldsymbol{\nu}_d^{\mathsf{T}} - \Big[\Big(\sum_{k=1}^{V} x_{ki}\Big)\frac{1}{R}\sum_{r=1}^{R}\theta_{dir}(\boldsymbol{\nu}_d^{\mathsf{T}} + \boldsymbol{\epsilon}_{dr}^{\mathsf{T}} \odot \exp\{\boldsymbol{\varsigma}^{\mathsf{T}}\})\Big]\Bigg]. \tag{5.30}$$

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

$$\tilde{\nabla}_{t_{ik}}\mathcal{L} \triangleq \begin{cases} \nabla_{t_{ik}}\mathcal{L} + \omega, & t_{ik} = 0, \;\; \nabla t_{ik} < -\omega \\ \nabla_{t_{ik}}\mathcal{L} - \omega, & t_{ik} = 0, \;\; \nabla t_{ik} > \omega \\ 0, & t_{ik} = 0, \;\; |\nabla t_{ik}| \leq \omega \\ \nabla_{t_{ik}}\mathcal{L}, & |t_{ik}| > 0 \quad . \end{cases} \tag{5.31}$$

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

$$\boldsymbol{t}_i \leftarrow \mathcal{P}_{\mathcal{O}}(\boldsymbol{t}_i + \boldsymbol{d}_i) \tag{5.32}$$

Algorithm 3: Stochastic VB training for Bayesian SMM

initialize the model and the variational parameters
repeat
  for $d = 1 \dots D$ **do**
    sample $\tilde{\boldsymbol{\epsilon}}_{dr} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \qquad r = 1 \dots R$
    compute $\mathcal{L}(q_d)$ using (5.24)
    compute gradient $\nabla \boldsymbol{\nu}_d$ using (5.27)
    compute gradient $\nabla \boldsymbol{\varsigma}_d$ using (5.29)
    update $\boldsymbol{\nu}_d$ and $\boldsymbol{\varsigma}_d$ using ADAM
  end
  compute $\mathcal{L}$ using (5.25)
  compute sub-gradients $\tilde{\nabla} \boldsymbol{t}_i$ using (5.30) and (5.31)
  update rows in $\boldsymbol{T}$ using (5.32)
until *convergence or max_iterations*

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

$$\boldsymbol{d}_i = \eta \, \mathrm{diag}(\sqrt{\hat{\boldsymbol{s}}_i} + \epsilon)^{-1} \hat{\boldsymbol{f}}_i \,. \tag{5.33}$$

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

$$\mathcal{P}_{\mathcal{O}}(\boldsymbol{t}_i + \boldsymbol{d}_i) \triangleq \begin{cases} 0 & \text{if} \quad t_{ik}(t_{ik} + d_{ik}) < 0, \\ t_{ik} + d_{ik} & \text{otherwise} \,. \end{cases} \tag{5.34}$$

The orthant projection introduces explicit zeros in the estimated $\boldsymbol{T}$ matrix and, results in sparse solution. The stochastic VB training is outlined in Algorithm 3.

5.4 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 (5.24). 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 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

Table 5.1: 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

5.5 Experimental details

5.5.1 Datasets

The experiments were conducted on both speech and text corpora. The speech data used is *Fisher* phase 1 corpus[2], 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 5.1 presents the details of data splits; they are the same as used in earlier research (Hazen et al., 2007; Hazen, 2011; May et al., 2015). See Appendix F for more insights into the dataset.

Our preprocessing involved removing punctuation and special characters, but we did not remove any stop words. Using Kaldi open-source toolkit (Povey et al., 2011b), we trained a sequence discriminative DNN-HMM automatic speech recognizer (ASR) system (Veselý et al., 2013a) 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*[3], 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. Additionally, the perplexity results are also reported with a limited vocabulary of 2000 words as used in (Srivastava et al., 2013b; Miao et al., 2016). This version of the corpus is not the pre-processed version as used in earlier chapter (§ 4.4.1).

[2] https://catalog.ldc.upenn.edu/LDC2004S13
[3] http://qwone.com/~jason/20Newsgroups/

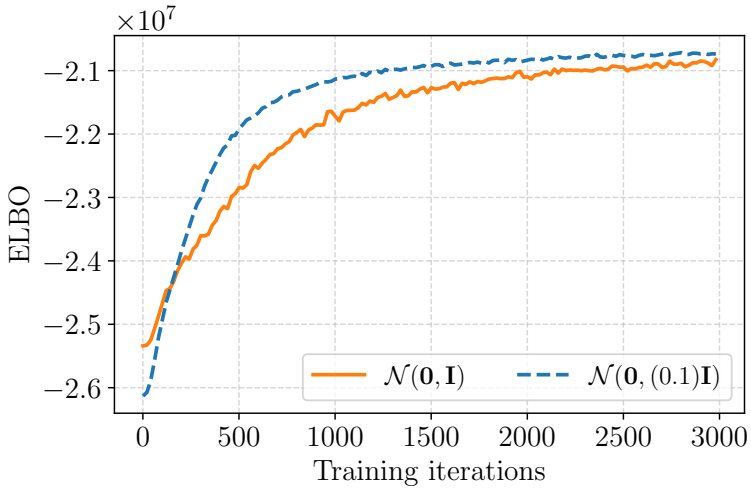


Figure 5.3: Convergence of Bayesian SMM for various initializations of variational distribution. The model was trained on *20Newsgroups* corpus with $K = 100$, and $\omega = 1$.

5.5.2 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 $\mathcal{N}(\mathbf{0}, \text{diag}(0.1))$ to speed up the convergence. Fig. 5.3 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 $\mathcal{N}(\mathbf{0}, \text{diag}(0.1))$ converges faster as compared to the one initialized to standard normal. In all the further experiments, we initialized both the prior and variational distributions to $\mathcal{N}(\mathbf{0}, \text{diag}(0.1))$. One can introduce hyper-priors and learn the parameters of prior distribution.

5.5.3 Evaluation using perplexity

These experiments show the comparison of the perplexities of proposed Bayesian SMM and neural variational document model (NVDM) (Miao et al., 2016). NVDM was reported to achieve state-of-the-art perplexity results on *20Newsgroups* dataset under limited vocabulary condition of 2000 words. Additionally, the comparison of perplexity results on *Fisher* corpora is also presented.

The publicly available source code for NVDM[4] was used and custom implementation[5] of Bayesian SMM using `PyTorch` (Paszke et al., 2017) library.

[4] https://github.com/ysmiao/nvdm
[5] https://github.com/skesiraju/BaySMM

Table 5.2: Comparison of perplexity (PPL) results on *20Newsgroups*. The values in the brackets indicate results with a limited vocabulary of 2000 words.

Model	**Embedding dimension** (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)

All the models were evaluated by measuring perplexity of the test documents. It is computed as an average document perplexity according to (2.9) and also computed across the entire test corpus according to (2.10). Perplexity gives a notion of how well the model explains (fits) the data or how uncertain the model is about the data. Lower perplexity values indicate that the model is less uncertain about the data. Perplexity is inversely proportional to log-likelihood of the data. This can be seen from (2.9) and (2.10).

In our case, $\ln p(\boldsymbol{x})$ from (5.11) 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 (5.5). We can only compute $\mathcal{L}(q)$, which is a lower bound on $\ln p(\boldsymbol{x})$; thus the resulting perplexity values act as upper bounds. This is true for NVDM (Miao et al., 2016) or any other model in the VB framework where the true posterior is intractable (Bishop, 2006). We estimated $\mathcal{L}(q)$ from (5.24) using 32 samples, i.e., $R = 32$, in order to compute perplexity. We used the same number of samples for the baseline NVDM. Fig. 5.4 shows the perplexity values of both the datasets evaluated using Bayesian SMM with various number of Monte Carlo samples R. We can observe that higher number of samples ($R \geq 16$) results in consistent perplexities.

Next, Table 5.2 presents the comparison of *20Newsgroups* test data perplexities obtained using Bayesian SMM and NVDM in. It also shows the perplexities of *20Newsgroups* corpus under full and a limited vocabulary of 2000 words, similar to the ones reported in (Miao et al., 2016). The Table 5.2 also shows the perplexities computed using the maximum likelihood probabilities estimated on the test data. It acts as the lower bound on the test perplexities.

NVDM was shown (Miao et al., 2016) to achieve superior perplexity scores when compared to LDA, docNADE (Larochelle and Lauly, 2012), Deep Auto Regressive Neural Network models (Mnih and Gregor, 2014). To the best of our knowledge, our model achieves state-of-the-art perplexity scores on *20Newsgroups* corpus under limited and full vocabulary conditions.

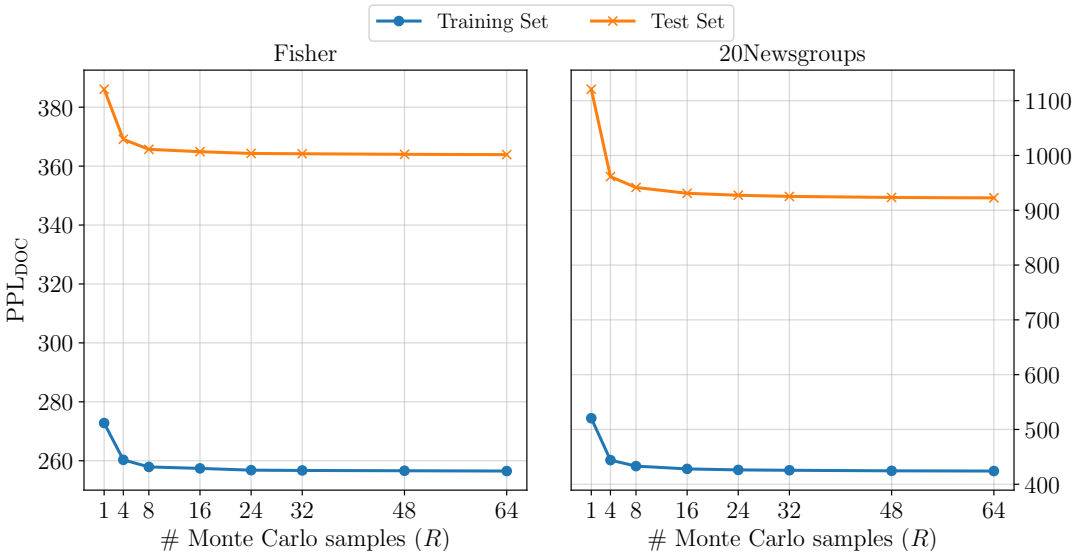
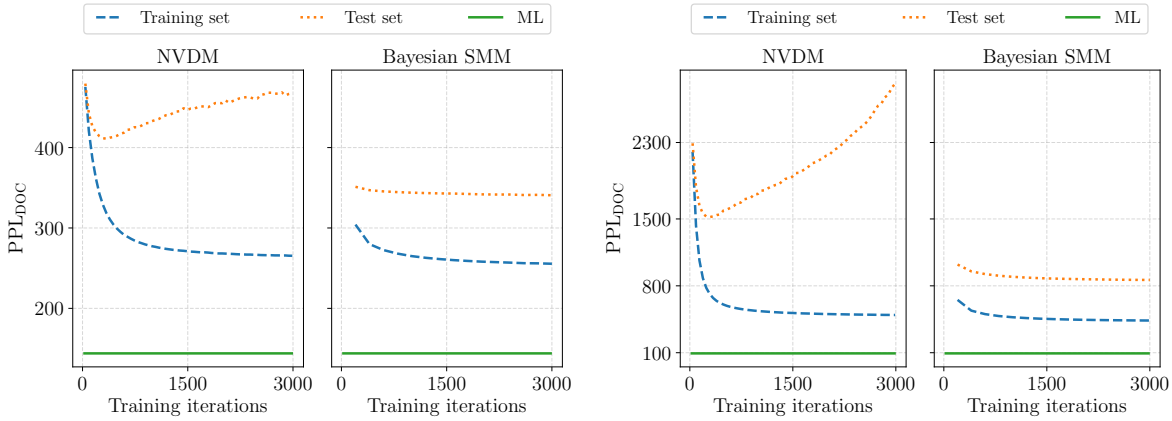


Figure 5.4: Perplexity values of both the datasets evaluated using Bayesian SMM with various number of Monte Carlo samples R.



(a) PPL of *Fisher* test data.

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

Figure 5.5: 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.

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 Figures 5.5a and 5.5b. 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 certain

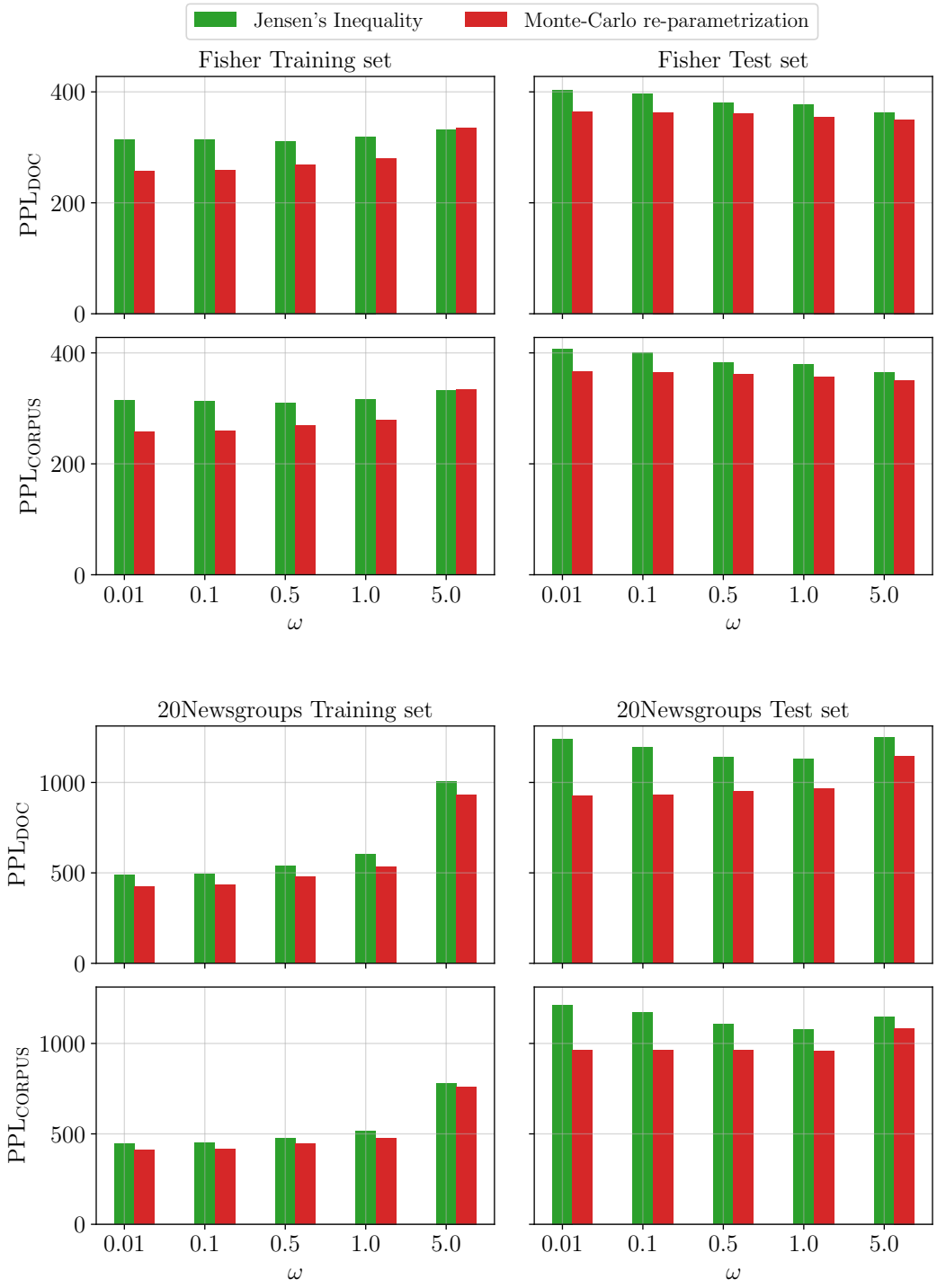


Figure 5.6: Comparison of perplexities for Bayesian SMM on two different datasets with two different bounds (Jensen's inequality and Monte Carlo re-parametrization). ω represents ℓ_1 regularization weight on the rows of matrix $\boldsymbol{T}$.

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.

5.5.4 Jensen's inequality vs re-parametrization trick

This section presents the comparison of bounds used in obtaining the ELBO for Bayesian SMM, i.e., we compare the Jensen's inequality with the Monte Carlo approximation via the re-parametrization trick (§ 5.2.1 and 5.2.2). In both cases, ELBO was optimized to learn the model parameters and also the posterior distribution over latent variables (document embeddings). We compare the perplexity scores of both the models on both the datasets under various hyper-parameter settings. Lower perplexities indicate a better fit to the data.

Recall from § 5.2.1 that by using Jensen's inequality in the formulation of ELBO, we only have a lower bound during the optimization. However, for the computation of perplexity, we use the same Monte-Carlo approximation for the expectation over log-sum-exp (5.23). This is fair because we are not estimating the parameters and only evaluating the perplexity. Moreover, we have also shown in Fig. 5.4 that higher number of Monte Carlo samples yields in robust estimates of perplexities.

The Fig. 5.6 present the perplexities on both training and test sets of both *Fisher* and *20Newsgroups* datasets. The models are compared for various choices of ω, the ℓ_1 regularization weight for the rows in matrix $\boldsymbol{T}$. The embedding dimension was set to 100 and all the other configurations were identical for both the models.

5.5.5 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. 5.7 shows an example of uncertainty captured in the embeddings. Here, the Bayesian SMM was trained on *20Newsgroups* with an embedding dimension of 100.

5.6 Summary and conclusions

In this chapter, we have presented Bayesian subspace multinomial model for learning document representations along with their uncertainties. We have also addressed the problem of intractability that appears when performing variational inference in mixed-logit models. We also presented the comparison of bounds for approximating the intractable expectation over log-sum-exp. The experiments revealed that Monte Carlo approximation via the re-parametrization

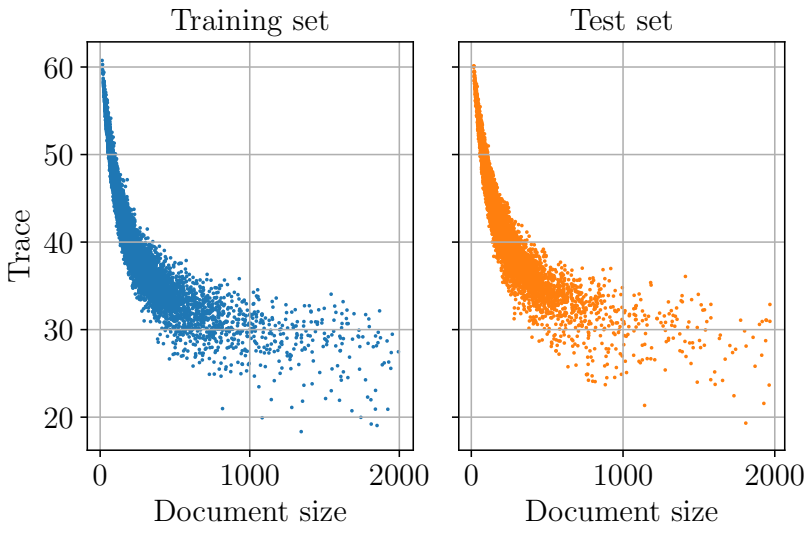


Figure 5.7: Uncertainty (trace of covariance of posterior distribution) captured in the document embeddings of *20Newsgroups* dataset.

trick is better than Jensen's inequality. The re-parametrization trick can be used in Bayesian modelling of word embedding algorithms, thereby capturing the uncertainties.

The experimental results have shown that the proposed model achieves state-of-the-art perplexities on *20Newsgroups* and *Fisher* test sets. Further, we have also illustrated the robustness of Bayesian SMM as compared to the variational auto encoder inspired document models.

The next chapter will present a classifier that can exploit the learned uncertainties for topic identification.

Chapter 6

Exploiting uncertainties in document embeddings for topic identification

This chapter will present a generative Gaussian classifier that exploits the uncertainty in the posterior distributions of document embeddings. Moreover, it also exploits the same uncertainty while predicting the class labels. More specifically, it will be used for the task of topic identification from spoken and text documents. The proposed classifier is called Gaussian classifier with uncertainty (GLCU) and is inspired by (Kenny et al., 2013; Cumani et al., 2015). It can be seen as a extension to the simple Gaussian linear classifier (GLC) (Bishop, 2006).

6.1 Gaussian linear classifier with uncertainty

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

The 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 $\boldsymbol{\mu}_\ell \in \mathbb{R}^K$ representing a column. GLC is described by the following linear model:

$$\boldsymbol{w}_d = \boldsymbol{\mu}_d + \boldsymbol{\varepsilon}_d, \tag{6.1}$$

where $\boldsymbol{\mu}_d = \boldsymbol{M}\boldsymbol{h}_d$, and $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \boldsymbol{0}, \boldsymbol{D}^{-1})$.

GLC can be trained by estimating the parameters $\Theta = \{\boldsymbol{M}, \boldsymbol{D}\}$ that maximize the class conditional likelihood of the training data. For a single training example, the likelihood is computed as:

$$p(\boldsymbol{w}_d \,|\, \boldsymbol{h}_d, \Theta) = \mathcal{N}(\boldsymbol{w}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{D}^{-1}). \tag{6.2}$$

In the standard scenario, GLC is trained using the observed document embeddings $\boldsymbol{w}_d$. 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 would aim to maximize the expected class-conditional likelihood, where the expec-

tation over $\boldsymbol{w}_d$ would be calculated for each training example with respect to the posterior distribution $q(\boldsymbol{w}_d)$:

$$\mathbb{E}_q[p(\boldsymbol{w}_d \,|\, \boldsymbol{h}_d, \Theta)] = \mathbb{E}_q[\mathcal{N}(\boldsymbol{w}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{D}^{-1})]. \tag{6.3}$$

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 latent variable $\boldsymbol{y}_d$ as:

$$\boldsymbol{\nu}_d = \boldsymbol{\mu}_d + \boldsymbol{y}_d + \boldsymbol{\varepsilon}_d, \tag{6.4}$$

where $p(\boldsymbol{y}_d) = \mathcal{N}(\boldsymbol{y}_d \,|\, \boldsymbol{0}, \boldsymbol{\Gamma}_d^{-1})$. The resulting model is called Gaussian linear classifier with uncertainty (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 (Bishop, 2006):

$$p(\boldsymbol{\nu}_d \,|\, \boldsymbol{h}_d, \Theta) = \mathcal{N}(\boldsymbol{\nu}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{\Gamma}_d^{-1} + \boldsymbol{D}^{-1}). \tag{6.5}$$

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 (6.2) vs (6.5).

GLCU can be trained by estimating its parameters Θ that maximize the class conditional likelihood Eq. (6.5) of training data. This can be done efficiently by using the `EM` algorithm; described in the following section.

6.1.1 EM algorithm

To estimate the model parameters, we iterate between `E-step` and `M-step`. In the `E-step`, we calculate the posterior distribution of latent variables[1]:

$$p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d, \Theta) \propto p(\boldsymbol{\nu}_d \,|\, \boldsymbol{y}_d, \Theta)\; p(\boldsymbol{y}_d) \tag{6.6}$$

$$\propto \mathcal{N}(\boldsymbol{y}_d \,|\, \boldsymbol{u}_d, \boldsymbol{V}_d^{-1}), \tag{6.7}$$

where

$$\text{mean} \quad \boldsymbol{u}_d = [\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{\Gamma}_d]^{-1}(\boldsymbol{\nu}_d - \boldsymbol{\mu}_d), \tag{6.8}$$

$$\text{and precision matrix} \quad \boldsymbol{V}_d = \boldsymbol{D} + \boldsymbol{\Gamma}_d. \tag{6.9}$$

[1] See Appendix C for complete derivation.

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

$$\mathcal{Q} = \mathbb{E}_p[\sum_{d=1}^{D} \log p(\boldsymbol{\nu}_d, \boldsymbol{y}_d \mid \Theta)] \tag{6.10}$$

$$= -\frac{1}{2}\Bigg[\sum_{d=1}^{D}\Big(\mathrm{tr}(\boldsymbol{D}\boldsymbol{V}_d^{-1}) + \boldsymbol{a}_d^{\mathsf{T}}\,\boldsymbol{D}\,\boldsymbol{a}_d\Big) - N\log|\boldsymbol{D}|\Bigg] + \text{const}, \tag{6.11}$$

where

$$\boldsymbol{a}_d = [\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d)]. \tag{6.12}$$

Maximizing the auxiliary function $\mathcal{Q}$ w.r.t. Θ, we have the following closed form update equations:

$$\boldsymbol{\mu}_\ell \leftarrow \frac{1}{|\mathcal{I}_\ell|}\sum_{d\in\mathcal{I}_\ell}(\boldsymbol{\nu}_d - \boldsymbol{u}_d) \quad \forall\, \ell = 1 \ldots L \tag{6.13}$$

$$\boldsymbol{D}^{-1} \leftarrow \frac{1}{N}\Big[\sum_{d=1}^{D}(\boldsymbol{a}_d\,\boldsymbol{a}_d^{\mathsf{T}}) + \boldsymbol{V}_d^{-1}\Big], \tag{6.14}$$

where $\mathcal{I}_\ell$ is the set of documents from class ℓ. To train the GLCU model, we alternate between E-step and M-step until convergence. For faster convergence, the model parameters $\{\boldsymbol{M}, \boldsymbol{D}\}$ are initialized with maximum likelihood estimates obtained from GLC.

6.1.2 Classification

Given a posterior distribution of a test document embedding $q(\boldsymbol{w}_t) = \mathcal{N}(\boldsymbol{w}_t \,|\, \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t^{-1})$, we compute the class conditional likelihood according to Eq. (6.5), and the posterior probability of a class $\mathcal{C}_k$ is obtained by applying the Bayes' rule:

$$p(\mathcal{C}_k \,|\, \boldsymbol{\nu}_t, \boldsymbol{\Gamma}_t, \Theta) = \frac{p(\boldsymbol{\nu}_t \mid \boldsymbol{\mu}_k, \boldsymbol{D}, \boldsymbol{\Gamma}_t)\, p(\mathcal{C}_k)}{\sum_\ell p(\boldsymbol{\nu}_t \mid \boldsymbol{\mu}_\ell, \boldsymbol{D}, \boldsymbol{\Gamma}_t)\, p(\mathcal{C}_\ell)} \tag{6.15}$$

6.2 Illustration using synthetic data

This section illustrates the significance of the proposed GLCU on synthetic data. To begin with, the procedure for generating data points along with their uncertainties is explained. Each data point is in the form a Gaussian distribution with a specific mean and covariance.

First, 4 Gaussian distributed classes with specific mean ($\star$) and a shared co-variance matrix are initialized. Then 100 data points are sampled for each class. A few of the samples are depicted in subplot (i.a) of Fig. 6.1. The next subplot (i.b) shows the estimated parameters of GLC, using the generated 100 points. The estimated class means are represented by ♦ and the

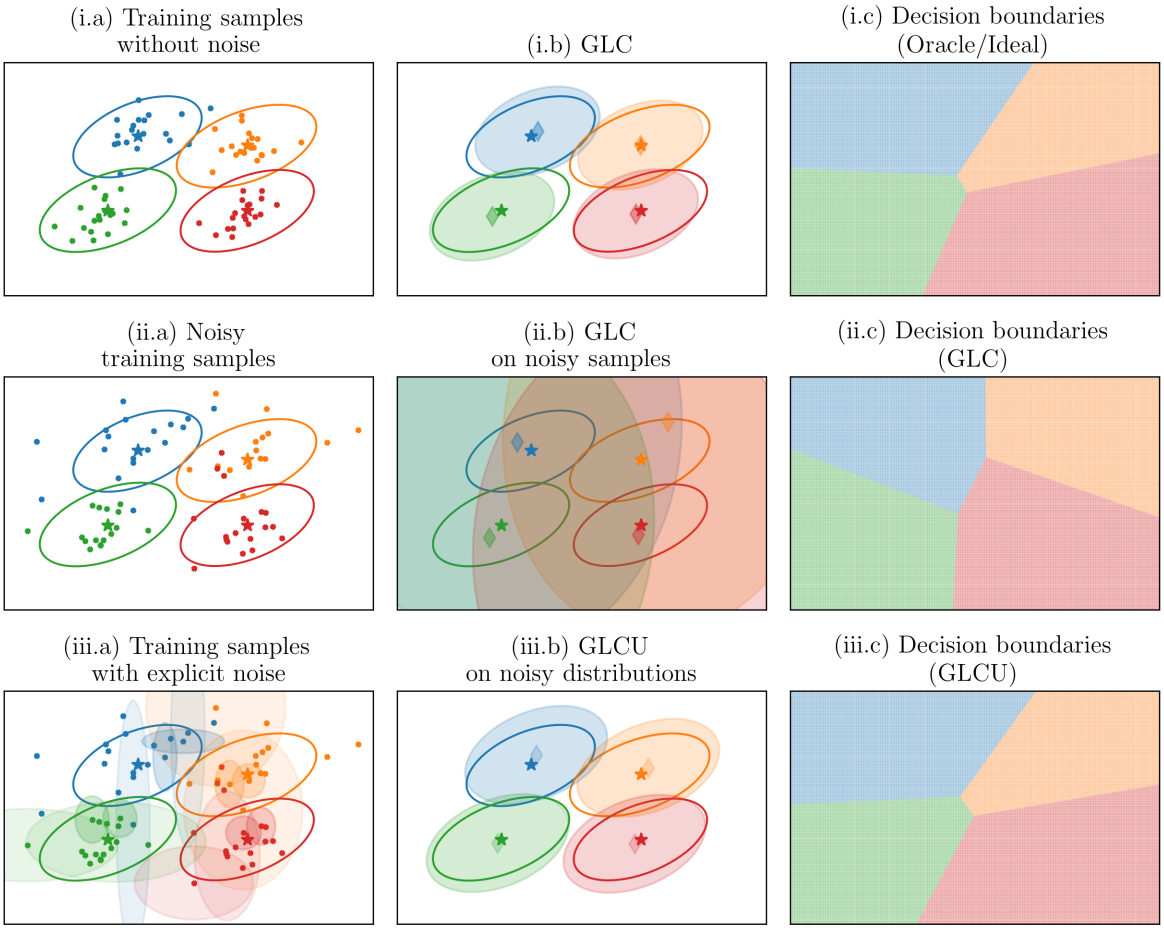


Figure 6.1: The illustration of GLC vs GLCU on two-dimensional synthetic data. The image should be read row-wise first and then compared column-wise. Refer to the text for details.

shared co-variance by an shaded ellipse. The corresponding linear decision boundaries are in subplot (i.c). Given that the data points are directly generated from true classes the estimated parameters and decision boundaries reflect the distribution of true classes.

Next, each data point x_i is corrupted by a random noise; sampled from Gaussian distribution having mean x_i and a specific precision λ_i (inverse of variance); which in turn is sampled from Gamma distribution. If x_i is the original data point, then the noisy data point $\hat{x}_i$ is generated as:

$$\lambda_i \sim \text{Gamma}(k, \theta), \tag{6.16}$$

$$\hat{x}_i \sim \mathcal{N}(x_i, \lambda_i^{-1}) \tag{6.17}$$

where k, θ are shape and scale parameters of Gamma distribution respectively.

Few noisy data points ($\hat{x}_i$) are illustrated in subplot (ii.a) in Fig. 6.1. The next subplot

(ii.b) shows the estimated parameters of GLC using the 100 noisy points. The corresponding decision boundaries are shown in subplot (ii.c). Note that estimated parameters and the decision boundaries are different from the true ones (i.b) and (i.c).

If we do not observe the true data points, the uncertainty associated with every noisy data point $\hat{x}_i$ is fully described by its precision λ_i. The subplot (iii.a) from Fig. 6.1 shows few noisy data points along with their uncertainties. The next subplot (iii.b) estimated class means and shared covariance using GLCU. We can observe that by estimated parameters (class means and shared-covariance) in (iii.b) are much closer to the true ones (i.b). Similarly the decision boundaries in (iii.c) resemble the true ones in (i.c).

The analogy to the posterior distribution of document embeddings (extracted using Bayesian SMM) is straightforward - $p(x_i) = \mathcal{N}(x_i \mid \hat{x}_i, \lambda_i^{-1})$, i.e., $\hat{x}_i$ and λ_i represent the estimated mean and precision of the Gaussian distribution. The inherent assumption in Bayesian SMM is that the inferred uncertainty is same as the true uncertainty. In reality, this may not be entirely true. Nevertheless, our results show the benefits of modelling and exploiting uncertainties.

6.3 Related works: modelling uncertainties via Gaussian embeddings

Recent works in NLP (Vilnis and McCallum, 2015; Sun et al., 2018) represent word embeddings in the form of Gaussian distributions. Using the asymmetric KL divergence or the symmetric Wasserstein Distance, the uncertainty is exploited for word similarity, entailment and document classification tasks. Similar to the presented classifier, (Xiao and Wang, 2019) quantifies the uncertainties in the data and exploits it for sentiment analysis, named entity recognition, etc.

Gaussian embeddings extracted from spoken utterance, popularly known as embeddings (Dehak et al., 2011) were used for speaker identification, and verification tasks; and have been the state-of-the-art for several years (Kenny et al., 2013). Ondel et al. (2019) proposed a fully Bayesian subspace hidden Markov model for acoustic unit discovery from speech; where phone-like (acoustic) units from an unseen language are represented by Gaussian embeddings living in a subspace that was learnt using labelled data from other languages. Brümmer et al. (2018) developed[2] a more theoretical framework around Gaussian embeddings for various classification and verification scenarios.

Kendall and Gal (2017) argued the importance of modelling uncertainty of safety critical applications in computer vision, and applied it for semantic segmentation and depth regression tasks.

[2] https://github.com/bsxfan/meta-embeddings

6.4 Experiments

In these experiments, the learned document representations are evaluated on topic identification task. The experiments are conducted on the same datasets as described earlier in 5.5.1.

6.4.1 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. It is valid for SMM, neural variational document model (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 6.1. GLC and GLCU does not have any hyper-parameters to tune, while the ℓ_2 regularization weight of LR was tuned using cross-validation experiments.

6.4.2 Baseline topic ID systems

6.4.2.1 NVDM

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 early stopping mechanism as described in 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 (Miao et al., 2016), 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 hyper-parameters were tuned based on cross-validation experiments.

6.4.2.2 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 hyper-parameters such as embedding dimension $K = \{100, \ldots, 800\}$, and regularization weight $\omega = \{1e-04, \ldots, 1e+01\}$. 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 6.5.1 shows that Bayesian SMM is more robust to over-fitting when compared to SMM and NVDM, and does not require an early stopping mechanism.

6.4.2.3 ULMFiT

The third baseline system is the universal language model fine-tuned for classification (ULMFiT) (Howard and Ruder, 2018). The pre-trained[3] 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: $\{0.2, \ldots, 0.6\}$. Fine-tuning LM for higher number of epochs degraded the classification performance. 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 hyper-parameters (drop-out rates, and fine-tuning epochs).

6.4.2.4 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 (May et al., 2015). The hyper-parameter (ℓ_2 regularization weight) of LR was selected based on 5-fold cross-validation experiments on training set.

6.5 Results and discussion

6.5.1 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 Sections 6.4.1, and 6.4.2, an early stopping mechanism (ESM) during the training of an unsupervised model (e.g.: NVDM, SMM, and Bayesian SMM) is required to get optimal

[3] https://github.com/fastai/fastai

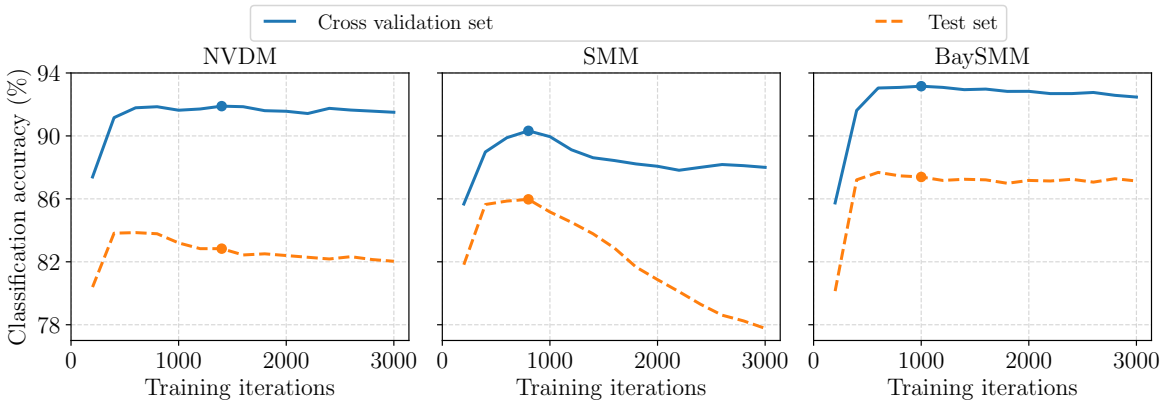


Figure 6.2: Performance of topic ID systems on *Fisher* data at various checkpoints during model training. The circular dot (•) represents the best cross-validation score and the corresponding test score obtained using the early stopping mechanism (ESM). The embedding dimension was set to 100 for all the models.

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[4] and test sets. Fig. 6.2 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 case of (non-Bayesian) SMM, the test accuracy drops significantly after 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 much lower than that of Bayesian SMM and also 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 in addition, does not require any ESM. In all the further topic ID experiments, we report classification results for Bayesian SMM without ESM; while the results for SMM, and NVDM are with ESM.

6.5.2 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

[4] 5-fold cross-validation on training set.

Table 6.1: Comparison of results on *Fisher* test sets, from earlier published works, our baselines and proposed systems. ⋆ indicates a pure discriminative model.

			Transcriptions			
			Manual		Automatic	
Systems	**Model**	**Classifier**	**Acc. (%)**	**CE**	**Acc. (%)**	**CE**
Prior works	BoW (Hazen et al., 2007)	NB	87.61	-	-	-
(Baselines)	TF-IDF (May et al., 2015)	LR	86.41	-	-	-
Baselines	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**

is about its prediction. A well calibrated classifier tends to have lower cross-entropy.

Table 6.1 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 et al., 2007), 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., 2015) had slightly larger vocabulary than ours, and their best system is similar to our baseline TF-IDF based system. The remaining rows in Table 6.1 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. It can be observed ULMFiT has the lowest cross-entropy among all the systems.

Table 6.2 presents classification results on *20Newsgroups* dataset. The first three rows give the results as reported in earlier works. (Pappagari et al., 2018), proposed a CNN-based discriminative model trained to jointly optimize categorical cross-entropy loss for classification

Table 6.2: 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 (Pappagari et al., 2018) *	-	**86.12**	-
	SCDV (Mekala et al., 2017)	SVM	**84.60**	-
	NTSG-1 (Liu et al., 2015)	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

task along with binary cross-entropy for verification task. Sparse composite document vector (SCDV) (Mekala et al., 2017) exploits pre-trained word embeddings to obtain sparse document embeddings, whereas neural tensor skip-gram model (NTSG) (Liu et al., 2015) extends the idea of a skip-gram model for obtaining document embeddings. The authors in (SCDV) (Mekala et al., 2017) have shown superior classification results as compared to paragraph vector, LDA, NTSG, and other systems. The next rows in Table 6.2 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 purely discriminative CNN model. We can also see that all the topic ID systems based on Bayesian SMM are consistently better than variational auto encoder inspired NVDM, and (non-Bayesian) SMM.

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 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 or model can be directly used for classification tasks without any fine-tuning.

6.6 Summary and conclusions

This chapter presented a generative Gaussian linear classifier (GLCU) that exploits the distribution of data points i.e., uncertainty in embeddings or embeddings. On synthetic data, the problem (inaccurate estimates of class means) caused by uncertain features (embeddings) is illustrated. Further, a proper way of estimating the class means by using the proposed GLCU was discussed. Applying the proposed GLCU on the document embedding posterior distributions extracted from Bayesian SMM, achieved state-of-the-art classification results on both *Fisher* speech and *20Newsgroups* text corpora while considering unsupervised topic models.

Chapter 7

Multilingual document embeddings

A closed-set monolingual topic identification (ID) or document classification in resource-rich languages is usually done with the help of discriminative models such as end-to-end neural network classifiers (Yang et al., 2016a; Pappagari et al., 2018) or pre-trained language models fine-tuned for classification (Howard and Ruder, 2018). In case of cross-lingual topic ID, where target data has little or no labels, learning a common embedding space for multiple (say, L number of) languages is beneficial (Ammar et al., 2016; Schwenk and Li, 2018; Ruder et al., 2019). This common embedding space is learnt by exploiting parallel dictionary or parallel sentences among the L languages. Such a parallel data is not required to have topic labels. A classifier is then trained on the embeddings from a source (SRC) language (one from the L languages) that has topic labels. The same classifier is then and used to classify the embeddings extracted for test data, which can be from any of the L target (TAR) languages. The underlying assumption here is that the embeddings carry semantic concept(s), independent of language, enabling cross-lingual transferability (SRC $\rightarrow$ TAR). Hence, the reliability of this *scheme* solely depends on quality of the embedding space. Note that the amount of available training data could be limited and different from the parallel data, which is also the case for the experiments presented in this chapter.

This chapter presents an extension of Bayesian SMM to learn language-agnostic document embeddings by exploiting multilingual parallel data. The proposed model aims to learn a common low-dimensional subspace for document-specific unigram distributions from multiple languages. Moreover, the proposed model represents the document embeddings in the form of Gaussian distributions, thereby encoding the uncertainty in its covariance. The learned uncertainties are further propagated into a generative Gaussian linear classifier for zero-shot cross-lingual topic identification.

The experiments on 5-language subset of Reuters multi-lingual corpora (MLDoc) show that the proposed system outperforms (a) multilingual word embedding based (Multi-CCA), and (b) state-of-the-art neural machine translation inspired sequence-to-sequence bi-directional long short-term memory network (BiLSTM) based systems (Schwenk and Li, 2018), with significant

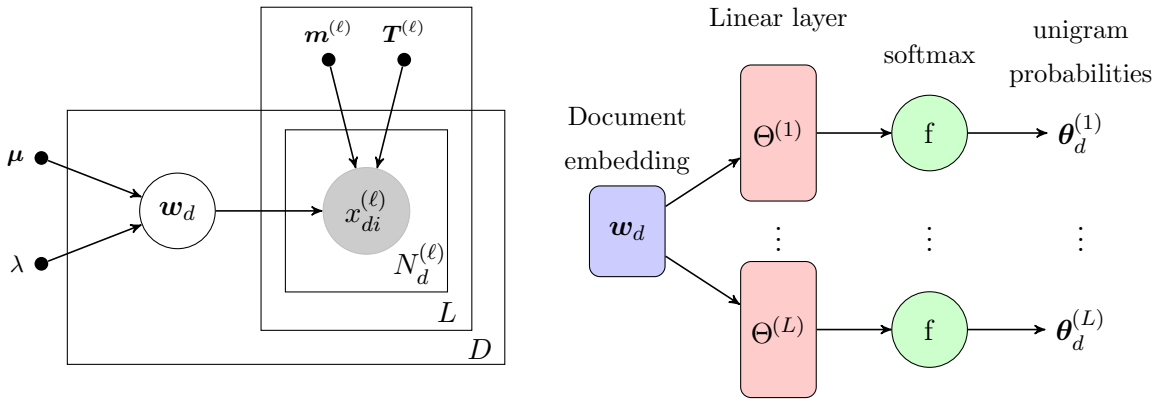


Figure 7.1: (Left) Graphical representation of the proposed multilingual model, where L represents number of languages and D denotes number of L-way parallel documents (translations). $\{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\}$ are document-independent, language-specific model parameters, whereas $\boldsymbol{w}_d$ is document-specific but language-independent random variable (embedding). $N_d^{(\ell)}$ represents number of word tokens in document d from language ℓ. (Right) Alternative representation, where document embedding $\boldsymbol{w}_d$ is a passed through language-specific linear layers whose parameters are $\Theta^{(\ell)} = \{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\}$. The outputs are sent through softmax function to obtain unigram distribution of words in document d for each language $\ell = 1 \dots L$.

margins in most of the transfer directions.

The experimental analysis also shows that increasing the amount of parallel data improves the overall performance of the cross-lingual topic ID systems. Nonetheless, exploiting the uncertainties during classification is always beneficial.

7.1 Model

The graphical representation of multilingual Bayesian subspace multinomial model is depicted in Fig. 7.1. Like majority of the probabilistic topic models (Blei, 2012; Miao et al., 2016), our model also relies on bag-of-words representation of documents. Let $V^{(\ell)}$ represent the vocabulary size in language $\ell = 1 \dots L$. Let $\{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\} \forall\, \ell$ represent the language-specific model parameters, where $\boldsymbol{T}^{(\ell)}$ is a low-rank matrix of size $V^{(\ell)} \times K$ ($K \ll V^{(\ell)}$) defines the subspace of document specific unigram distributions. Our multilingual model assumes that the L-way parallel data (translations of bag-of-words) are generated according to the following process:

First, sample a K-dimensional ($K \ll V^{(\ell)}$) language-independent, document-specific embedding from isotropic Gaussian prior distribution with precision λ:

$$\boldsymbol{w}_d \sim \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, (\lambda \boldsymbol{I})^{-1}). \tag{7.1}$$

$\boldsymbol{w}_d$ can be interpreted as vector representing higher-level semantic concepts (topic alike) of a document, independent of any language. For each language $\ell = 1 \dots L$, a vector of word counts $\boldsymbol{x}_d^{(\ell)}$ is generated by the following two steps:

(i) Compute the document-specific unigram distribution $\boldsymbol{\theta}_d^{(\ell)}$ using the language-specific parameters:

$$\boldsymbol{\theta}_d^{(\ell)} = \text{softmax}(\boldsymbol{m}^{(\ell)} + \boldsymbol{T}^{(\ell)}\, \boldsymbol{w}_d), \tag{7.2}$$

(ii) Sample a vector of word counts $\boldsymbol{x}_d^{(\ell)}$:

$$\boldsymbol{x}_d^{(\ell)} \sim \text{Multinomial}(\boldsymbol{\theta}_d^{(\ell)},\, N_d^{(\ell)}), \tag{7.3}$$

where $N_d^{(\ell)}$ are the number of trials (word tokens in document d), i.e., $\sum_n x_{dn}^{(\ell)} = N_d^{(\ell)}$.

$\boldsymbol{x}^{(1)} \dots \boldsymbol{x}^{(L)}$ represent L-way parallel bag-of-words statistics.

The above steps describe the generative process of the proposed multilingual topic model. However, in reality, we do not generate any data, instead we invert the generative process: given the training (observed) data $\boldsymbol{x}_d^{(\ell)}\, \forall\, \ell = 1 \dots L$, $\forall\, d = 1 \dots D$, we estimate the language-specific model parameters $\{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\}$ and also the posterior distributions of language-independent document embeddings $p(\boldsymbol{w}_d | \boldsymbol{x}_d^{(1)} \dots \boldsymbol{x}_d^{(L)})\, \forall\, d$. Moreover, given an unseen document $\boldsymbol{x}_t^{(\ell)}$ from any of the L languages, we infer the corresponding posterior distribution of the document embedding $p(\boldsymbol{w}_t \mid \boldsymbol{x}_t^{(\ell)})$. Note that such a posterior distribution also carries the uncertainty about the estimate.

Although we describe the model assuming L-way parallel data, in practice the model can be trained with parallel text (translations) between language pairs covering all the L languages.

7.1.1 Variational Bayes training

The proposed model is trained using the variational Bayes framework, i.e., we approximate the intractable true posterior with the variational distribution:

$$q(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{\nu}_d,\, \text{diag}(\boldsymbol{\gamma}_d)^{-1}), \tag{7.4}$$

and, optimize the evidence lower-bound. Further, we use Monte Carlo samples via the re-parametrization trick to approximate the expectation over log-sum-exp term which appears in the lower-bound (see Chapter 5 § 5.2.2 for details). The resulting lower-bound for a single set of L-parallel documents in given by:

$$\mathcal{L}(q_d) \approx \sum_{\ell=1}^{L} \sum_{i=1}^{V^{(\ell)}} x_{di}^{(\ell)} \Big[(m_i^{(\ell)} + \boldsymbol{t}_i^{(\ell)}\, \boldsymbol{\nu}_d) - \frac{1}{R} \sum_{r=1}^{R} \log \Big(\sum_{j=1}^{V} \exp\{m_j^{(\ell)} + \boldsymbol{t}_j^{(\ell)}\, g(\boldsymbol{\epsilon}_{dr})\} \Big) \Big] - D_{\text{KL}}(q_d \,||\, p), \tag{7.5}$$

where $D_{\mathrm{KL}}(q_d \,||\, p)$ is the Kullback-Leibler divergence from variational distribution (7.4) to the prior (7.1) and, $g(\boldsymbol{\epsilon}_{dr}) = \boldsymbol{\nu} + \boldsymbol{\gamma} \odot \tilde{\boldsymbol{\epsilon}}_{dr}$, with $\tilde{\boldsymbol{\epsilon}}_{dr} \sim \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \boldsymbol{I})$. R are the number of Monte Carlo samples used for empirically approximating the expectation over log-sum-exp. The derivation of the lower-bound for a monolingual case is given in § 5.2.

The complete lower-bound is just the summation over all the documents. Additionally, we use ℓ_2 regularization term with weight ω for language-specific model parameters $\{\boldsymbol{T}^{(\ell)}\} \forall\, \ell$. Thus, the final objective is

$$\mathcal{L} = \sum_{d=1}^{D} \mathcal{L}(q_d) \; - \; \omega \sum_{\ell=1}^{L} \sum_{i=1}^{V^{(\ell)}} || \, \boldsymbol{t}_i^{(\ell)} \, ||_2 \, . \tag{7.6}$$

In practice, we follow batch-wise stochastic optimization of (7.6) using ADAM (Kingma and Ba, 2015). In each iteration, we update the all model parameters $\{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\} \forall\, \ell$ and the corresponding posterior distributions of document embeddings $q(\boldsymbol{w}_d) \forall\, d$.

Unlike in earlier chapters, we use ℓ_2 regularization here, because the optimization is easier when performing batch-wise training on a large dataset. ℓ_1 regularization with orthant-wise learning leads poor minima while performing batch-wise updates, since the objective function is estimated on a batch of data rather than on the entire dataset. Further more, the orthant projection introduces explicit zeros, which makes the batch-wise training even more difficult.

7.1.2 Extracting embeddings for unseen documents

Given a bag-of-word statistics from an unseen document from any of the L languages, we can infer (extract) the corresponding document embedding along with its uncertainty. This is done by keeping the language-specific model parameters $\{\boldsymbol{m}^{(\ell)}, \boldsymbol{T}^{(\ell)}\}$ fixed, and iteratively optimizing the objective in (7.5) with respect to the parameters of the variational distribution. In the resulting $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\nu}, \mathrm{diag}(\boldsymbol{\gamma})^{-1})$, the mean $\boldsymbol{\nu}$ represents the (most likely) document embedding, and variance $\mathrm{diag}(\boldsymbol{\gamma})^{-1}$ encodes the uncertainty around the mean $\boldsymbol{\nu}$.

7.2 Classification exploiting uncertainties

In a traditional scenario, where we have only point estimates of embeddings, all the embeddings are considered equally important by a classifier. This may not be true all the time. For example, shorter and ambiguous documents can result in poor estimates of the embeddings, which can affect the classifier during training and the performance during prediction. Since our proposed model yields document embeddings represented by Gaussian distributions, with the uncertainty about the embedding encoded in the covariance, we use two linear classifiers that can exploit this uncertainty. The first one is the generative Gaussian linear classifier with uncertainty (GLCU) 6 The second one is the discriminative multi-class logistic regression with uncertainty (MCLRU).

7.2.1 Generative classifier

In general, for any classification task, we estimate the posterior probability of class label ($\mathcal{C}_k$) given a feature vector (embedding) $\boldsymbol{w}$

$$p(\mathcal{C}_k \mid \boldsymbol{w}) = \frac{p_\theta(\boldsymbol{w} \mid \mathcal{C}_k)\, p(\mathcal{C}_k)}{\sum_j p_\theta(\boldsymbol{w} \mid \mathcal{C}_j)\, p(\mathcal{C}_j)} \tag{7.7}$$

where, $p_\theta(\boldsymbol{w} \mid \mathcal{C}_k)$ is the likelihood function parametrized by θ, and $p(\mathcal{C}_k)$ is the class prior. In case of generative classifiers, the likelihood function is assumed to have a known parametric form (e.g. Gaussian, Multinomial).

For Gaussian linear classifier (GLC), the likelihood function is $p_\theta(\boldsymbol{w} \mid \mathcal{C}_k) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\mu}_k, \boldsymbol{S}^{-1})$, where $\boldsymbol{w}$ is the input feature (point estimate of the embedding), $\boldsymbol{\mu}_k$ is the mean of class $\mathcal{C}_k$, and $\boldsymbol{S}$ is the precision matrix shared across all the classes.

Given that the our input features (embeddings) come in the form of Gaussian distributions, i.e., $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\nu}, \text{diag}(\boldsymbol{\gamma})^{-1})$, we can integrate out (exploit) the uncertainty in the input while evaluating the likelihood function. In case of generative Gaussian classifier, where the likelihood function is also Gaussian, the expected likelihood has an analytical form

$$p_\theta(\boldsymbol{\nu} \mid \mathcal{C}_k) = \mathbb{E}_q[p_\theta(\boldsymbol{w} \mid \mathcal{C}_k)] = \mathcal{N}(\boldsymbol{\nu} \mid \boldsymbol{\mu}_k, \boldsymbol{S}^{-1} + \text{diag}(\boldsymbol{\gamma})^{-1}). \tag{7.8}$$

GLC with likelihood function replaced by (7.8) is called GLCU. Both are essentially the same classifiers, i.e., they have the same assumptions about the underlying data and hence the same model parameters. The only difference lies in the evaluation of likelihood function.

7.2.2 Discriminative classifier

For discriminative classifier such as multi-class logistic regression (MCLR), the posterior probability of class label ($\mathcal{C}_k$) given an input feature vector $\boldsymbol{w}$ is

$$p(\mathcal{C}_k \mid \boldsymbol{w}) = \frac{\exp\{\boldsymbol{h}_k^\top \boldsymbol{w} + b_k\}}{\sum_j \exp\{\boldsymbol{h}_j^\top \boldsymbol{w} + b_j\}}, \tag{7.9}$$

where $\{b_k, \boldsymbol{h}_k\} \forall\, k$ are the parameters of the classifier. Unlike in GLC, we cannot analytically compute the expectation over (7.9) with-respect-to the input features (Gaussian distributions). Instead we approximate the expectation using Monte Carlo samples (Kendall and Gal, 2017; Xiao and Wang, 2019):

$$p(\mathcal{C}_k \mid \boldsymbol{w}) = \mathbb{E}_q\Big[\frac{\exp\{\boldsymbol{h}_k^\top \boldsymbol{w} + b_k\}}{\sum_j \exp\{\boldsymbol{h}_j^\top \boldsymbol{w} + b_j\}}\Big] \approx \frac{1}{M}\sum_{m=1}^{M} \frac{\exp\{\boldsymbol{h}_k^\top \boldsymbol{\varepsilon}_m + b_k\}}{\sum_j \exp\{\boldsymbol{h}_j^\top \boldsymbol{\varepsilon}_m + b_j\}}, \quad \boldsymbol{\varepsilon}_m \sim q(\boldsymbol{w}) \,\forall\, m. \tag{7.10}$$

Eq. (7.10) represents the posterior probability computation for MCLRU.

Theoretically, given the true uncertainties in the training examples, GLCU and MCLRU can better estimate the model parameters of the classifier. Similarly, it can also exploit the uncertainties in the test examples during classification.

However, in our case, the uncertainties are estimated using our Bayesian multilingual topic model as described in § 7.1.2. The underlying assumption here is that uncertainties extracted using our model are close enough to the true uncertainties as expected by the classifiers. This assumption is empirically supported through our experimental results presented in § 7.5.

7.3 Related works

7.3.1 Multilingual embeddings in NLP

Multilingualism in machine learning models can be achieved using word embeddings, or joint sentence (document) embeddings or pre-trained language models sharing a common vocabulary and/or parameters.

Ammar et al. (2016) used canonical correlation analysis (CCA) to map word embeddings from several languages to a common space. These mapped embeddings are used in a convolutional neural network for cross-lingual topic ID Schwenk and Li (2018).

Using parallel data from Europarl, Schwenk and Li (2018) trained a sequence-to-sequence (seq2seq) model comprising of BiLSTM layers to learn a common embedding space for sentences from multiple languages. In their model, each language has a separate encoder and decoder. A similar seq2seq model was used by Artetxe and Schwenk (2019), with a shared byte-pair-encoded vocabulary over 93 languages. The encoder is BiLSTM with 5 layers, where as the decoder is a single LSTM layer, which additionally takes language ID (embedding) as input. Embeddings for new test data are obtained by forward propagating through the encoder. This is followed by a two hidden layered feed-forward neural network classifier for cross-lingual topic ID.

Almost all of the recent models relating to multilingual learning with cross-lingual transfer applications rely on massive amounts of data for pre-training, followed by fine-tuning all the parameters or only the embedding layer (Artetxe et al., 2020). BERT (Devlin et al., 2019) is a transformer based pre-trained language model. Multi-lingual BERT (mBERT) (Wu and Dredze, 2019) uses shared word piece vocabulary from 104 languages and aims to learn cross-lingual representations without any parallel data. On the other hand multilingual translation encoder (MMTE) (Siddhant et al., 2019) uses the transformer architecture for neural machine translation, whose encoder is fine tuned for classification tasks.

	Min. sentence length constraint				
	30	**35**	**40***	**45**	**50**
Language	**Vocabulary size** ($V^{(\ell)} \times 1000$)				
English (EN)	36	33	30	27	23
German (DE)	84	72	61	51	42
French (FR)	46	42	37	33	28
Italian (IT)	56	50	44	39	33
Spanish (ES)	56	51	45	39	34
# sentences	1.6M	1.1M	0.73M*	0.48M	0.31M

Table 7.1: Data statistics under various sentence length constraints. * indicates the data on which hyper-parameters are tuned.

7.4 Experimental setup

7.4.1 Datasets

Europarl (v7) contains numerous parallel sentences between several European language pairs (Koehn, 2005). We considered 5 languages namely, English (EN), German (DE), French (FR), Italian (IT) and Spanish (ES) and constructed 5-way parallel sentences. Using English as reference, we retained sentences that are at least 40 words in length; which resulted in 146K 5-way parallel sentences. The maximum number of sentences are 146k×5 = 0.73M. In reality, not every sentence has a translation in all 5 languages. Later in § 7.5.2, we present the comparison of our systems with various amounts of parallel data, that are obtained by varying the sentence length constraint in the set $\{30, 35, 40, 45, 50\}$.

MLDoc (Reuters multilingual corpus vols 1, and 2) (Lewis et al., 2004) is a collection of more than 800k news stories in 14 languages[1], written by local news reporters. The news stories were manually classified into 4 topics, namely CCAT (Corporate/ Industrial), ECAT (Economics), GCAT (Government/Social) and MCAT (Markets). Using the standardized data preparation framework (Schwenk and Li, 2018), we created 5 class-balanced splits, where each split has 1000 training, 1000 development and 4000 test documents. We report the average classification accuracy of the 5 splits.

[1]English, Dutch, French, German, Chinese, Japanese, Russian, Portuguese, Spanish, Latin American Spanish, Italian, Danish, Norwegian, and Swedish.

Hyper-parameters		
Multilingual model	K	$\{50, 100, 200, 256\}$
	ω	$\{1e-4, 5e-3, \ldots, 1e-1\}$
MCLR	α	$\{1e-4, 5e-3, \ldots, 1e+2\}$

Table 7.2: Model hyper-parameters, where K is the embedding dimension, ω and α are the ℓ_2 regularization weights for the multilingual model and MLCR respectively.

7.4.2 Pre-processing

The vocabulary was built using only the multi-aligned Europarl corpus. Table 7.1 presents the vocabulary statistics. All the words were lower-cased and punctuation was stripped. Further, words that do not occur in at least two sentences were removed.

7.4.3 Hyper-parameters and model configurations

The proposed Bayesian multilingual topic model has 2 important hyper-parameters, i.e., latent (embedding) dimension K and ℓ_2 regularization weight ω corresponding to the model parameters $\{\boldsymbol{T}^{(\ell)}\} \forall\, \ell$. Table 7.2 presents the list of hyper-parameters we explored in our experiments. The prior distribution (7.1) was set to $\mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, (0.1\boldsymbol{I}))$ and the variational distribution (7.4) was initialized to be the same as prior. This enabled us to same learning rate for both mean and variance parameters. A batch size of 4096 was used during training. A constant learning rate of 0.05 was used both during training and inference. The model is trained for 2000 epochs and inference is done for 2000 iterations to obtain the posterior distributions.

The Gaussian linear classifier with uncertainty (GLCU) has no hyper-parameters to tune. We added ℓ_2 regularization term with weight α (Table 7.2) for the parameters of multi-class logistic regression (MCLR). The classifier was trained for a maximum 100 epochs using ADAM with a constant learning rate of $5e-2$. For multi-class logistic regression with uncertainty (MCLRU), we used $M = 32$ for the empirical approximation (7.10). $M > 32$ did not affect the classification performance significantly but, lower values degraded the performance for about 5%. Our models are implemented using PyTorch Paszke et al. (2017).

7.4.4 Proposed topic ID systems

The two linear classifiers GLC and MCLR use only the point estimates of the embeddings, i.e., they cannot exploit uncertainty during training and test. In the experiments we used only the mean parameter ($\boldsymbol{\nu}$) as the point estimate of document embedding. Contrastingly, GLCU and MCLRU are trained with the full posterior distribution $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\nu}, \text{diag}(\boldsymbol{\gamma})^{-1})$.

Transfer type	Model training	Model selection	Evaluation
Zero-shot	Train L_1	Dev L_1	Test L_i
Targeted	Train L_1	Dev L_2	Test L_2
Joint	Train L_i	Dev L_i	Test L_i

Table 7.3: Different schemes of cross and multilingual document classification (Schwenk and Li, 2018). Zero-shot transfer experiments are reported in this thesis.

7.4.5 Baseline systems

Our baseline systems for comparison are based on multilingual word embeddings + CNN classifier (MULTI-CCA) and BiLSTM based seq2seq models (Schwenk and Li, 2018). We denote BILSTM-EU (Schwenk and Li, 2018) as the system trained on 5 European languages similar to our systems.

Further, we also compare with the seq2seq BiLSTM trained on 93 languages sharing a common encoder (Artetxe and Schwenk, 2019). We represent this as BILSTM-93. Since the published work (Artetxe and Schwenk, 2019) only reports results from EN $\rightarrow$ XX, we took the full matrix of results from the corresponding github repository maintained by the authors[2]. These are the improved results since the publication. BILSTM-93 was trained on 16 NVIDIA V100 GPUs which took about 5 days (Artetxe and Schwenk, 2019).

Although all of these models use the same MLDoc corpus for cross-lingual topic ID, the multi-lingual embedding models are trained on different amounts of data comprising of various languages, hence we cannot directly compare all the models. However, we can compare BiLSTM-EU with our PRIMARY SYSTEM, since both models use the same 5 European languages from Europarl.

7.5 Results and discussion

We present full matrix of results, i.e., all possible training-test combinations among the 5 languages. It shows the cross-lingual performance in all transfer directions, enabling a detailed understanding. Fig. 7.2 shows accuracy on the development for various regularization weights ω. We split the results into two parts: *in language* represents same source and target language pair, where as *zero-shot transfer* implies different source and target language pairs. Note that MCLR performs best on *in language* setting, whereas GLCU and MCLRU perform the best in *zero-shot transfer* setting. However, model selection was based only on the *in language* performance. For MCLRU, $K = 256, \omega = 5e-3$ was found to give best results on the development set (in language average = 88.12). Similarly, for GLCU, $K = 256, \omega = 1e-3$ was found to

[2] https://github.com/facebookresearch/LASER/tree/master/tasks/mldoc

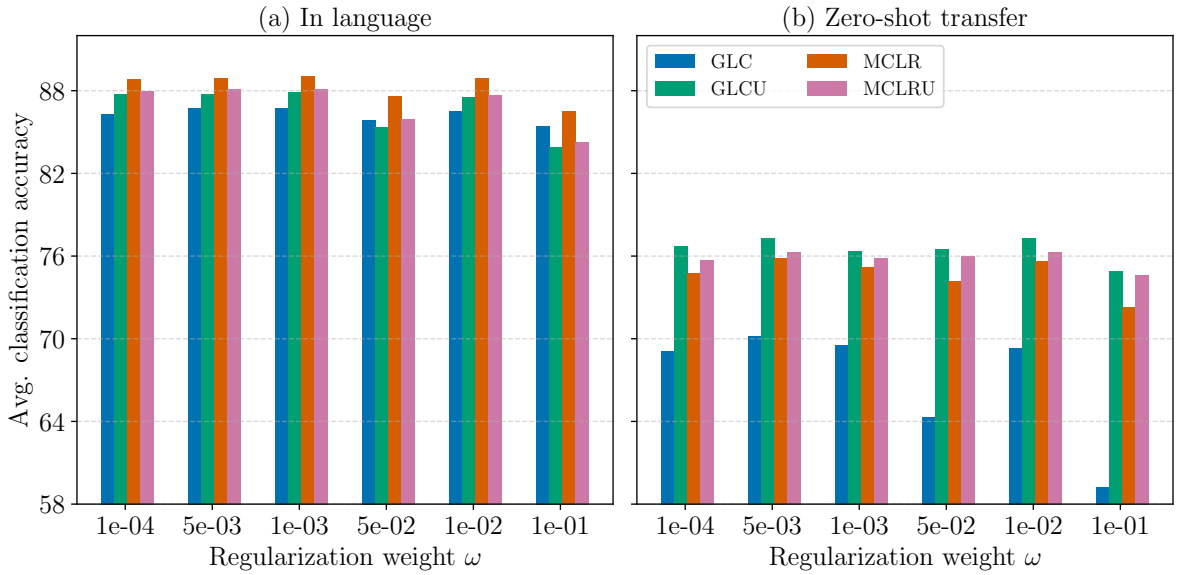


Figure 7.2: Comparison of average classification accuracies on dev set for various hyper-parameters (ω), and classifiers. The embedding dimension $K = 256$.

give best results on the development set (in language average = 87.91). These two are our PRIMARY SYSTEMS; each of which has about 56 million parameters and took about 22 hours to train on a single NVIDIA Tesla P-100 GPU. Since the language-specific model parameters are independent inferring the embeddings can be easily parallelized.

7.5.1 Zero-shot cross-lingual transfer

Table 7.4 presents the zero-shot classification results of our PRIMARY SYSTEM with GLCU and MCLRU respectively. These are the average accuracies from 5 test splits (§ 7.4.1). All the further comparisons are made with-respect-to these PRIMARY SYSTEMS.

Table 7.5 shows the absolute differences in classification accuracy between our PRIMARY SYSTEMS and each of the baseline systems. The positive bold value indicate the absolute improvement of our system as compared the respective baseline system. Note that the first two baseline systems are slightly better when training and test language are same, but significantly worse in transfer directions. This suggests that these models over-fit on the source language and generalizes poorly to the target languages.

As a specific example, by examining the results of MULTI-CCA (Table 4 from (Schwenk and Li, 2018), alternatively, we can infer the same in Table 7.5 in this Chapter), it can be observed that the system performs better when training and testing on the same language. Moreover Multi-CCA is slightly better when transferring from EN $\rightarrow$ XX, but relatively worse is other cases such as IT $\rightarrow$ XX, and XX $\rightarrow$ DE, suggesting a language bias in the embedding space.

	GLCU					MCLRU				
	Test language					Test language				
	EN	DE	FR	IT	ES	EN	DE	FR	IT	ES
EN	86.99	83.90	80.23	65.14	72.60	87.04	83.04	78.39	64.40	73.51
DE	74.04	91.25	81.75	63.50	76.79	74.61	91.67	82.45	66.67	76.97
FR	77.00	85.60	90.34	69.00	78.74	76.21	86.11	89.81	70.69	79.05
IT	71.89	79.36	80.22	80.89	79.69	71.63	80.56	80.37	80.93	79.23
ES	73.14	81.75	81.17	72.32	89.45	72.43	77.93	79.79	71.68	90.12

Table 7.4: Average test accuracies of the PRIMARY SYSTEMS with GLCU (Left) and MCLRU (Right).

Note that our PRIMARY SYSTEMS out performs Multi-CCA and BiLSTM-EU in majority of the transfer directions with significant margins, and more over performs competitively with the state-of-the-art BiLSTM-93 system. On an average, our PRIMARY SYSTEMS (GLCU, MCLRU) are 9.2% and 5.6% better than Multi-CCA and BiLSTM-EU respectively; and only 1.6% worse than BiLSTM-93 in the zero-shot cross-lingual transfer (off-diagonal). Note that BiLSTM-BIG is trained with 223M parallel sentences across 93 languages whereas our PRIMARY SYSTEM is trained on just 730k parallel sentences across 5 languages.

7.5.2 Significance of uncertainties in low-resource scenario

In this section, we compare the zero-shot topic ID performance of various classifiers with the embeddings extracted using our multilingual model. Given that we have only 1000 examples for training the classifiers, we can see the importance of modelling and utilizing uncertainties under such low-resource setting.

To better illustrate the importance of uncertainties, we trained GLC and MCLR with only the mean parameters, but during the test (prediction) time, we used the full posterior distributions (along with uncertainties) of the test document embeddings. This is valid because both GLC and GLCU have exactly the same model parameters (§ 7.2.1). Similarly MCLR and MCLRU are have exactly the same model parameters (§ 7.2.2). We represent these two classifiers as GLCU-P and MCLRUP, where -P denotes *uncertainty exploited only during prediction*.

The comparisons with GLCU-P and MCLRUP is presented in conjunction with the amount of parallel data that was used for training our multilingual embedding model. For simplicity, we present results in two parts, *in language* and *zero-shot transfer*. Figure 7.3 shows the average score on development set of all the 6 classifiers for varying amounts of parallel data. The overall performance of the systems increase slightly with the amount of parallel data. Nonetheless, exploiting the uncertainties, only even during the test time (GLCUP, MCLRUP) is always beneficial.

	GLCU					MCLRU				
	Test language					Test language				
	EN	DE	FR	IT	ES	EN	DE	FR	IT	ES
MULTI-CCA (Schwenk and Li, 2018)										
EN	-5.10	**1.42**	**6.17**	-5.62	**0.89**	-5.16	**1.84**	**6.01**	-4.98	**1.01**
DE	**17.30**	-2.24	**10.78**	**2.26**	**3.84**	**18.66**	-2.03	**10.90**	**2.69**	**3.74**
FR	**11.52**	**31.73**	-2.50	**8.74**	**13.25**	**11.41**	**32.41**	-2.69	**9.54**	**13.65**
IT	**17.34**	**28.76**	**16.99**	-4.28	**20.33**	**17.93**	**31.36**	**18.12**	-4.62	**20.55**
ES	-1.67	**23.00**	**13.66**	**12.88**	-4.53	-1.57	**22.13**	**14.16**	**13.33**	-4.32
BILSTM-EU (Schwenk and Li, 2018)										
EN	-1.30	**10.80**	**5.75**	**3.03**	**6.74**	-1.36	**11.21**	**5.59**	**3.67**	**6.86**
DE	**1.73**	-0.57	**6.88**	**9.79**	**1.57**	**3.09**	-0.36	**7.00**	**10.22**	**1.47**
FR	**0.32**	**7.01**	**0.25**	**6.19**	**7.95**	**0.21**	**7.69**	**0.06**	**6.99**	**8.35**
IT	**3.89**	**11.74**	**14.17**	-1.61	**11.94**	**4.48**	**14.34**	**15.30**	-1.95	**12.16**
ES	**9.63**	**7.75**	**16.62**	**13.30**	**1.64**	**9.73**	**6.88**	**17.12**	**13.75**	**1.85**
BILSTM-93 (Artetxe and Schwenk, 2019)										
EN	-3.74	-2.35	**2.20**	-5.06	-6.70	-3.69	-3.21	**0.36**	-5.80	-5.79
DE	-6.71	-1.45	-1.08	-9.75	-2.81	-6.14	-1.03	-0.38	-6.58	-2.63
FR	-3.08	-1.43	-0.46	-2.08	**0.34**	-3.87	-0.92	-0.99	-0.39	**0.65**
IT	-2.26	-1.37	**1.87**	-5.04	-2.91	-2.52	-0.17	**2.02**	-5.00	-3.37
ES	**3.56**	**2.02**	**5.87**	**1.22**	**0.70**	**2.85**	-1.80	**4.49**	**0.58**	**1.38**

Table 7.5: Comparison of our PRIMARY SYSTEMS (GLCU (Left) and MCLRU (Right)) with the baseline systems. Bold value indicates absolute improvement of our system over the respective baseline.

7.5.3 Results for reference

In Table 7.6, we present the cross-lingual topic ID results from the recently published works for reference. Note that all the systems were evaluated on MLDoc corpus, but the multilingual representation (embedding) model was trained on different amounts of data from various languages. Only BILSTM-EU and our PRIMARY SYSTEM are trained on the Europarl corpus with the same 5 languages. Moreover MBERT and BILSTM-93 are models with relatively huge number of parameters which take enormous computational resources to train; whereas our model can be trained under a day on a single GPU.

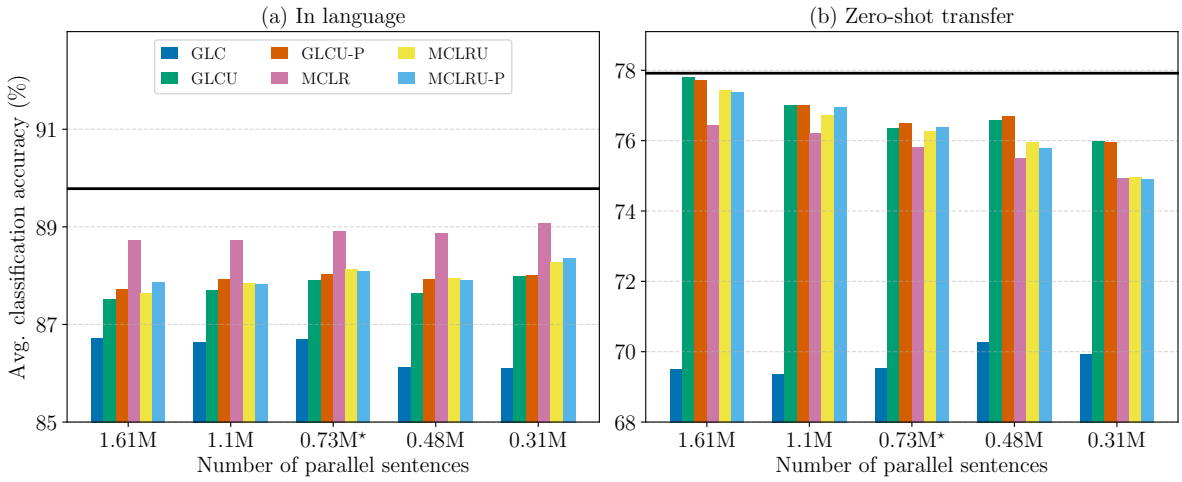


Figure 7.3: Comparison of average classification accuracies for various classifiers and varying amounts of parallel data. Model trained with 0.73M parallel sentences was the PRIMARY SYSTEM. The horizontal black line indicates the performance of BILSTM-93.

System	Number of languages in training data	Test language				
		EN	DE	FR	IT	ES
MBERT (Wu and Dredze, 2019)	104	<u>94.20</u>	80.20	72.60	68.90	72.60
MMTE (Siddhant et al., 2019)	103	**94.70**	77.40	77.20	64.20	73.00
BILSTM-93 (Artetxe and Schwenk, 2019)	93	90.73	**86.25**	78.03	**70.20**	**79.30**
MULTI-CCA (Schwenk and Li, 2018)	5	92.20	81.20	72.38	<u>69.38</u>	72.50
BILSTM-EU (Schwenk and Douze, 2017)	5	88.40	71.83	72.80	60.73	66.65
PRIMARY SYSTEM (GLCU)	5	86.99	<u>83.90</u>	**80.23**	65.14	72.60
PRIMARY SYSTEM (MCLRU)	5	87.04	83.04	<u>78.39</u>	64.40	<u>73.51</u>

Table 7.6: Results of multi-lingual zero-shot topic ID systems from EN → XX. Bold and underline indicates the first and second best scores respectively.

7.5.4 Topic discovery

To further understand our multilingual model, we took the point estimates (mean parameter, $\boldsymbol{\nu}$) of document embeddings of all the 5 languages from test set of MLDoc corpus, and clustered using k-means with 10 clusters. We took cluster centroids ($\bar{\boldsymbol{c}}_k$) of the 4 most dense clusters and projected these vectors on to the individual language specific subspaces $\{\boldsymbol{T}^{(\ell)}\}\,\forall \ell = 1 \ldots 5$

$$\boldsymbol{\theta}_k^{(\ell)} = \boldsymbol{T}^{(\ell)} \bar{\boldsymbol{c}}_k\,, \quad \forall \ell = 1 \ldots 5,\ \forall k = 1 \ldots 4 \tag{7.11}$$

The magnitude of values in $\boldsymbol{\theta}_k^{(\ell)} \in \mathbb{R}^{V^{(\ell)}}$ indicates the significance (representativeness) of the words from language ℓ to the cluster k. Table 7.7 presents top 4 words from each language for

each of the 4 clusters. Note that we did not use any parallel dictionary in our model, yet we can discover semantically related words across multiple languages.

EN	tyrant, gorostiaga, authoritarianism, tribal
DE	friedlicher (more peaceful), friedliebenden (peace-loving), kriegsverbrecher (war criminal), anfuhrern (lead)
FR	colonel (colonel), pacifiquement (peacefully), gorostiaga (*proper n.*), tyran (tyrant)
IT	pacifiste (pacifist), sradicato (uprooted), miloseviæ (*proper n.*), tribali (tribal)
ES	tirano (tyrant), vil (vile), magrebi (*n. North-west Africa*), tribales (tribal)
EN	inflation, inflationary, predictions, slowdown
DE	wirtschaftsindikatoren (econimic indicators), haushaltsdefiziten (budget deficts), inflationsrate (inflation rate), wirtschaftsdaten (economic data)
FR	inflationniste (inflationary), inflation, inflationnistes (inflationary), pronostics (prediction)
IT	inflazione (inflation), inflazionistici (inflationary), inflazionistiche (inflationary), ciclica (cyclical)
ES	inflacion (inflation), inflacionistas (inflationists), predicciones (predictions), coyuntural (conjunctural)
EN	overvaluation, yen, lira, dollar
DE	dollars ($), yuan (¥), wechselkurses (exchange rate), chinesischem (chinese)
FR	surevaluation (over valuation), croissent (grow), dollar ($), degonflement (deflating)
IT	sopravvalutazione (over estimation), valutari (currency), yen (¥), dollaro($)
ES	dolar ($), fly, yen (¥), redondeo (rounding)
EN	shareholding, artemis, aerospace, shareholder
DE	sesar (-), verwaltungsgesellschaften (management companies), double, verwaltungsgesellschaftspasses (management company passport)
FR	sesar (-), participations, guichet (counter), exportatrice (exporter)
IT	sesar (-), azionario (equity), neutralizzata (neutralized), double
ES	sesar (seize), financiara (will finance), accionarial (share holder), ccctb (Common Consolidated Corporate Tax Base)

Table 7.7: Top 4 representative words from each language for top 4 dense clusters obtained via k-means. English translations are given in parenthesis.

7.6 Conclusions

In this chapter, we presented a Bayesian multilingual topic model, which learns language-independent document embeddings along with their uncertainties. We propagated the uncertainties into a generative and discriminative linear classifier for zero-shot cross-lingual topic ID. Our systems out performed former state-of-the-art BiLSTM, and multilingual word embedding based system in majority of the transfer directions with significant margins. Moreover our systems perform competitively to the state-of-the-art universal sentence encoder, while only

requiring fraction of training data and computational resources. Our detailed experiment analysis emphasizes the importance of modelling and exploiting uncertainties for cross-lingual topic ID.

Chapter 8

Novel variants of Bayesian SMM

This chapter discusses novel variants of Bayesian SMM. We combine supervised and unsupervised objectives with-in a probabilistic framework that gives rise to newer models. Developing such models using a probabilistic framework gives an elegant interpretation to the model parameters and latent variables. Next, we discuss model to learn sentence embeddings by exploiting the contextual n-grams. This chapter only presents the theoretical details of the models. The experimental comparisons are left for future research.

8.1 Hybrid model

So far we have presented Bayesian SMM as a generative model for bag-of-words representation of documents. The model can be trained on largely available un-labelled data. The embeddings extracted from such unsupervised models may not be as competitive as pure discriminative model, when used for a supervised task such as document classification. On the other hand, discriminative models can only be trained on labelled data. Moreover, they require re-training (adaptation) to newer data and classes, which is computationally expensive.

However, it is possible to design (hybrid) models that can take advantage of both the labelled and unlabelled data (Lasserre, 2008), thus bridging the gap between generative and discriminative models. Given the background in Bayesian SMM, we present a hybrid variant below:

Let $\boldsymbol{X}_{\mathcal{U}}$ and $\boldsymbol{X}_{\mathcal{L}}$ represent bag-of-words statistics of un-labelled and labelled documents, comprising a vocabulary of size V. Let $\boldsymbol{Y}_{\mathcal{L}}$ represent the class labels corresponding to the labelled documents from $\boldsymbol{X}_{\mathcal{L}}$. Every row $\boldsymbol{x} \in \boldsymbol{X}_{\mathcal{U}} \cup \boldsymbol{X}_{\mathcal{L}}$ is a $1 \times V$ vector of word counts representing a single document, whereas every row $\boldsymbol{y} \in \boldsymbol{Y}_{\mathcal{L}}$ is a one-hot encoded vector of dimension $1 \times L$ representing the true class label for documents in $\boldsymbol{X}_{\mathcal{L}}$.

Let $\boldsymbol{\Theta}_{\mathcal{G}} = \{\boldsymbol{m}, \boldsymbol{T}\}$ and $\boldsymbol{\Theta}_{\mathcal{D}} = \{\boldsymbol{b}, \boldsymbol{H}\}$ represent the parameters of the hybrid model, where $\boldsymbol{\Theta}_{\mathcal{G}}$ and $\boldsymbol{\Theta}_{\mathcal{D}}$ are the parameters of the generative and discriminative parts respectively. The following steps describe the generative process of the training data, according to our hybrid

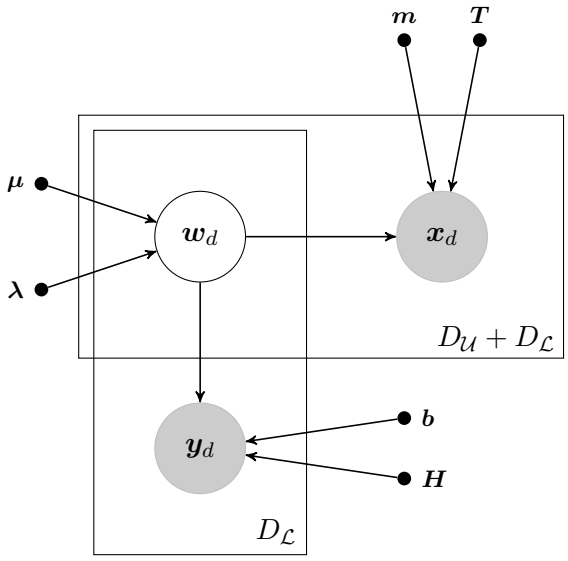


Figure 8.1: Graphical representation of the proposed hybrid model. $D_{\mathcal{U}}$ and $D_{\mathcal{L}}$ are the number of un-labelled and labelled documents respectively. $\boldsymbol{w}_d$ is the document-specific latent variable, $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$ are the observed variables. $\{\boldsymbol{m}, \boldsymbol{T}\}$ and $\{\boldsymbol{b}, \boldsymbol{H}\}$ are the model parameters specific to the generative and discriminative parts of the model respectively.

model:

For each document d in $\boldsymbol{X}_{\mathcal{U}} \cup \boldsymbol{X}_{\mathcal{L}}$, sample a document-specific latent variable:

$$\boldsymbol{w}_d \sim \mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{0}, \text{diag}(\lambda)^{-1}). \tag{8.1}$$

Generate vector of word counts for each document:

$$\boldsymbol{\phi}_d = \text{softmax}(\boldsymbol{m} + \boldsymbol{T}\,\boldsymbol{w}_d), \tag{8.2}$$

$$\boldsymbol{x}_d \sim \text{Multi}(\boldsymbol{\phi}_d; N_d). \tag{8.3}$$

Generate class labels only for documents in $\boldsymbol{Y}_{\mathcal{L}}$:

$$\boldsymbol{\varphi}_d = \text{softmax}(\boldsymbol{b} + \boldsymbol{H}\,\boldsymbol{w}_d), \tag{8.4}$$

$$\boldsymbol{y}_d \sim \text{Multi}(\boldsymbol{\varphi}_d; 1). \tag{8.5}$$

The above generative process fully describes the hybrid model. Now, given the training data $\{\boldsymbol{X}_{\mathcal{U}}, \boldsymbol{X}_{\mathcal{L}}, \boldsymbol{Y}_{\mathcal{L}}\}$, we would like to estimate the model parameters $\{\boldsymbol{\Theta}_{\mathcal{G}}, \boldsymbol{\Theta}_{\mathcal{D}}\}$ in addition to finding the posterior distribution of latent variables (document embeddings) $p(\boldsymbol{w}_d \mid \boldsymbol{x}_d, \boldsymbol{\Theta}_{\mathcal{G}})$ and posterior distribution of class labels given a document embedding $p(\boldsymbol{y}_d \mid \boldsymbol{w}_d, \boldsymbol{\Theta}_{\mathcal{D}})$.

The graphical model for the above generative process is illustrated in Fig. 8.1. From the graphical model, we can write the conditional independence as $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{y} \mid \boldsymbol{w}$. The variables $\boldsymbol{x}$ and

$\boldsymbol{y}$ are conditionally independent given $\boldsymbol{w}$:

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) = p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{w})\, p(\boldsymbol{w}) \tag{8.6}$$

$$= p(\boldsymbol{x} \mid \boldsymbol{w})\, p(\boldsymbol{y} \mid \boldsymbol{w})\, p(\boldsymbol{w}) \tag{8.7}$$

The conditional independence property is used in the further equations. We can write the posterior distribution of the latent variables as (explicit conditioning on model parameters is omitted for brevity):

$$p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{y}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{w}) p(\boldsymbol{y} \mid \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{x}) p(\boldsymbol{y})} \tag{8.8}$$

$$= \underbrace{\left[\frac{p(\boldsymbol{x} \mid \boldsymbol{w}) p(\boldsymbol{w})}{\int p(\boldsymbol{x} \mid \boldsymbol{w}) p(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}\right]}_{\mathsf{A}} \underbrace{\left[\frac{p(\boldsymbol{y} \mid \boldsymbol{w})}{\int p(\boldsymbol{y} \mid \boldsymbol{w}) p(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}\right]}_{\mathsf{B}}. \tag{8.9}$$

The numerator terms in (8.9) are given by:

$$p(\boldsymbol{x} \mid \boldsymbol{w}) = \prod_{i=1}^{V} \phi_i^{x_i} \tag{8.10}$$

$$= \left[\frac{\exp\{m_i + \boldsymbol{t}_i\, \boldsymbol{w}\}}{\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j\, \boldsymbol{w}\}}\right]^{x_i} \tag{8.11}$$

$$p(\boldsymbol{y} \mid \boldsymbol{w}) = \prod_{l=1}^{L} \varphi_l^{y_l} \tag{8.12}$$

$$= \left[\frac{\exp\{b_l + \boldsymbol{h}_l\, \boldsymbol{w}\}}{\sum_{j=1}^{L} \exp\{b_j + \boldsymbol{h}_j\, \boldsymbol{w}\}}\right]^{y_l}, \tag{8.13}$$

and $p(\boldsymbol{w})$ is given by (8.1).

In (8.9), term A is valid for the observed documents (both labelled and un-labelled), whereas term B is valid only for the observed class labels. Moreover, the denominator in both the terms cannot be evaluated because of the non-conjugacy. We can resort to variational inference as discussed in Chapter 5.2 to find the approximate to the true posterior by optimizing the evidence lower bound (ELBO).

During the VB training, in the E-step, the posterior distribution of latent variables is dependent on both the data and class labels. This enforces certain amount of discriminating nature in embedding space, which in turn influences estimation the generative model parameters $\boldsymbol{\Theta}_{\mathcal{G}}$ (M-step). During the inference (extraction), the model consists only the generative part.

8.2 Sentence embeddings exploiting contextual n-grams

All the models presented so far assume bag-of-words representation of documents, where the word-order is ignored. This simplified representation may not be optimal for every downstream

task. It is possible to design variants of Bayesian SMM that can exploit the contextual n-gram information. Subspace n-gram model (SnGM) is one such variant (Soufifar et al., 2013). In this section, we present a more generic version, where SnGM can be seen as a special case. We define the following terminology:

1. Let $x_a^{(s)}$ represent a word n-gram in a sentence s. We refer to this as an anchor n-gram or simply *anchor*. It can be a uni-gram or a bi-gram. In SnGM (Soufifar et al., 2013), the anchor is a bi-gram.

2. Let $\boldsymbol{c}_a^{(s)}$ represent the set of words in the *context* of the anchor $x_a^{(s)}$ in the sentence s. The *context* here can refer to the words succeeding, preceding or surrounding $x_a^{(s)}$ within a window of length L. Different choices can result in different models. SnGM models uni-gram distribution, succeeding the anchor bi-gram $x_a^{(s)}$. If the anchor is a uni-gram and the contextual window includes 2 words in each left and right contexts, we will have a model analogous to the skip-gram model (Mikolov et al., 2013).

Given a vocabulary of size V and the definitions of *anchor* and context (window) length L, we can have model parameters specific to each anchor, i.e., $\boldsymbol{\Theta}_a = \{\boldsymbol{m}_a, \boldsymbol{T}_a\} \quad \forall\, a = 1 \dots V$. Every sentence s is assumed to be generated by the following process:

Sample a K-dimensional ($K \ll V$) sentence-specific embedding $\boldsymbol{w}^{(s)}$:

$$\boldsymbol{w}^{(s)} \sim \mathcal{N}(\boldsymbol{w}^{(s)} \mid \boldsymbol{0}, \mathrm{diag}(\lambda)^{-1}). \tag{8.14}$$

The following steps are repeated until the desired sentence length is reached:

Sample an anchor:

$$x_a^{(s)} \sim \mathrm{Multi}(\boldsymbol{\alpha}, 1). \tag{8.15}$$

The probability of the contextual words given the anchor is obtained as:

$$\boldsymbol{\phi}_a^{(s)} = \mathrm{softmax}(\boldsymbol{m}_a + \boldsymbol{T}_a, \boldsymbol{w}^{(s)}). \tag{8.16}$$

Sample L number of contextual words:

$$\boldsymbol{c}_a^{(s)} \sim \mathrm{Multi}(\boldsymbol{\phi}_a^{(s)}, L). \tag{8.17}$$

$\boldsymbol{\alpha}$ in (8.15) refers to the uni-gram distribution of words in the vocabulary or the prior distribution over all the anchors. Note that anchor need not always be sampled from the uni-gram distribution (8.15). One can sample an anchor from the set of contextual words $\boldsymbol{c}_a^{(s)}$, enforcing autoregression.

If we constrain the anchor to be uni-gram and the window to be one word to the right, we will obtain a bi-gram language model:

$$p^{(s)}(x_1 x_2 \dots x_n) = \prod_{i=2}^{n} p^{(s)}(x_i \mid x_{i-1}) \tag{8.18}$$

where the distribution of next word, given previous (anchor) word is according to:

$$p^{(s)}(x_i \mid x_{i-1} = x_a) = \text{softmax}(\boldsymbol{m}_a + \boldsymbol{T}_a \boldsymbol{w}^{(s)}) \tag{8.19}$$

In the above, note that model parameters are shared across sentences, whereas latent variables are specific to sentences. The number of parameters for such a model will grow quadratically with respect the size of the vocabulary (or the number of anchors):

$$\#\text{ model parameters} = V \times ((V+1) \times K) \tag{8.20}$$

We can factorize of the parameter space to reduce the total number of parameters (Novotný et al., 2019).

Further, we can use the same variational inference framework with the approximation techniques on any of the above models.

8.3 Summary

This chapter presented few novel variants of Bayesian SMM. We have seen a hybrid model that can combine both the generative and discriminative modelling taking advantage of unlabelled and labelled data. Next we have also seen variants of Bayesian SMM that can learn sentence embeddings by exploiting the contextual n-grams. These are only few possible models, however one can combine the ideas from deep learning and variational autoencoders to devise many Bayesian models.

Chapter 9

Conclusions and directions for future research

This thesis presented novel methods for modelling text documents. Using a simple log-linear model called subspace multinomial model (SMM) for learning document embeddings, the experiments reported have shown that the obtained embeddings are superior as compared to classical topic models such as latent Dirichlet allocation, and sparse topical coding. The document embeddings extracted from SMM exhibit Gaussian-like distribution, which enabled us to simple Gaussian linear classifier and k-means clustering algorithms. With the help of ℓ_1 regularization over the parameters of SMM, and employing orthant-wise learning, sparsity is induced into the model, improving the generalization capabilities as compared to the ℓ_2 regularized model. A further analysis showed that the unsupervised topic models like SMM require an additional early stopping mechanism in order to yield embeddings optimal for downstream supervised tasks such as topic identification.

Next, using the variational Bayes framework, a novel extension to SMM called Bayesian SMM was proposed; which can represent document embeddings in the form of Gaussian distributions; thereby encoding the uncertainty in its covariance. Empirically, it was shown that the uncertainty captured in the covariance of the posterior Gaussian distributions is inversely proportional to document length, i.e., embeddings obtained from shorted document tend to me more uncertain as compared to the ones obtained from longer documents. The proposed Bayesian SMM achieved state-of-the-art perplexity results on *20Newsgroups* text and *Fisher* speech corpora, outperforming the variational autoencoder inspired document model, with a significant margin. Additionally, the thesis also addressed the problem of intractability (expectation over log-sum-exp) that commonly appears while performing variational inference in mixed-logit models. The experiments have shown that the approximation using the Monte Carlo samples via the re-parametrization trick is superior to the bounds obtained via Jensen's inequality. This scheme can be applied while performing Bayesian inference in language models, or word embedding methods or any mixed-logit model.

The learned embedding uncertainties from Bayesian SMM are further propagated into the proposed generative Gaussian linear classifier for topic identification. The topic ID experiments

have shown that the proposed system is robust to over-fitting and does not require any early stopping mechanism. The proposed system achieved state-of-the-art results on *Fisher* speech and *20Newsgroups* corpora as compared to other unsupervised topic and document models; and achieves comparable results to the supervised discriminative models (classifiers).

Further, the Bayesian SMM is extended to multilingual scenario for obtaining semantic-rich language independent document embeddings. Using the Gaussian linear classifier with uncertainty, the proposed system was used for zero-shot cross-lingual topic ID on *Europarl* and *Reuters multilingual news corpora*. On an average, our systems achieve 9.2% and 5.6% better zero-shot classification results as compared ton Multi-CCA (multilingual word embedding based) and BiLSTM based systems respectively. Our system is only 1.6% worse than BiLSTM-93 (trained on 93 languages) in the zero-shot cross-lingual transfer (off-diagonal). Note that BiLSTM-93 is trained with 223M parallel sentences across 93 languages, which takes about 5 days on 16 NVIDIA V-100 GPUS; whereas our Bayesian system was trained on just 730k parallel sentences across 5 languages one a single NVIDIA Tesla P-100 GPU, under a day. Further, experiment analysis has demonstrated that in a low-resource cross-lingual transfer scenario, learning and exploiting the uncertainties is beneficial irrespective of the amount data available for learning the common embedding space.

Towards the end, a few variants of Bayesian SMM are discussed theoretically (i) a hybrid variant that bridges the gap between generative and discriminative models, and (ii) a model that for learning sentence embeddings by exploiting the contextual n-grams.

Our future work involves exploring (deep) Bayesian models for language representation with applications to cross-lingual natural language inference, named-entity-recognition. We aim to train hybrid models, by exploiting both labelled and unlabelled data for natural language understanding tasks. Since the amounts of data collected and stored from various (including low-resource) languages of the world is increasing at an astronomical rate, we hope the models presented in this thesis foster research using Bayesian approaches.

Publications

Pre-prints

1. S. Kesiraju, S. Sagar, O. Glembek, L. Burget, and S. V. Gangashetty. A Bayesian multilingual document model for zero-shot cross-lingual topic identification, 2020b. URL https://arxiv.org/abs/2007.01359

Published articles

Journals

1. S. Kesiraju, O. Plchot, L. Burget, and S. V. Gangashetty. Learning document embeddings along with their uncertainties. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2319–2332, 2020a. doi: 10.1109/TASLP.2020.3012062

Conference proceedings

1. S. Kesiraju, L. Burget, I. Szöke, and J. Černocký. Learning Document Representations Using Subspace Multinomial Model. In *Proceedings of Interspeech, ISCA*, pages 700–704, September 2016. doi: 10.21437/Interspeech.2016-1634. URL http://dx.doi.org/10.21437/Interspeech.2016-1634

2. S. Kesiraju, R. Pappagari, L. Ondel, L. Burget, S. V. Gangashetty, et al. Topic identification of spoken documents using unsupervised acoustic unit discovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 5745–5749, March 2017. URL https://www.fit.vutbr.cz/research/groups/speech/publi/2017/kesiraju_icassp2017_0005745.pdf

3. S. Kesiraju, G. Mantena, and K. Prahallad. IIIT-H System for MediaEval 2014 QUESST. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain*. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1263/mediaeval2014_submission_76.pdf

4. K. Prahallad, A. Vadapalli, S. Kesiraju, H. A. Murthy, S. Lata, T. Nagarajan, M. Prasanna, H. Patil, A. K. Sao, S. King, A. W. Black, and K. Tokuda. The Blizzard Challenge 2014. In *Proceedings of Blizzard Workshop*, 2014

5. M. Hannemann, J. Trmal, L. Ondel, S. Kesiraju, and L. Burget. Bayesian joint-sequence models for grapheme-to-phoneme conversion. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 2836–2840, March 2017

6. 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 2018*, pages 3383–3387, 2018. doi: 10.21437/Interspeech.2018-1070. URL http://dx.doi.org/10.21437/Interspeech.2018-1070

7. L. Ondel, L. Burget, J. Černocký, and S. Kesiraju. Bayesian phonotactic language model for acoustic unit discovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 5750–5754, March 2017. URL http://www.fit.vutbr.cz/research/groups/speech/publi/2017/hannemann_icassp2017_0002836.pdf

8. B. Pulugundla, M. K. Baskar, S. Kesiraju, E. Egorova, M. Karafiát, L. Burget, and J. Černocký. BUT System for Low Resource Indian Language ASR. In *Proc. Interspeech 2018*, pages 3182–3186, 2018. doi: 10.21437/Interspeech.2018-1302. URL http://dx.doi.org/10.21437/Interspeech.2018-1302

9. O. Plchot, P. Matejka, O. Novotnỳ, S. Cumani, A. Lozano-Diez, J. Slavicek, M. Diez, F. Grézl, O. Glembek, K. Mounika, A. Silnova, L. Burget, L. Ondel, S. Kesiraju, and J. Rohdin. Analysis of BUT-PT Submission for NIST LRE 2017. In *Odyssey - The Speaker and Language Recognition Workshop, ISCA*, pages 47–53, June 2018

10. O. Plchot, P. Matějka, R. Fér, O. Glembek, O. Novotný, J. Pešán, K. Veselý, L. Ondel, M. Karafiát, F. Grézl, S. Kesiraju, L. Burget, N. Brummer, P. du Albert Swart, S. Cumani, H. S. Mallidi, and R. Li. BAT System Description for NIST LRE 2015. In *Odyssey - The Speaker and Language Recognition Workshop, ISCA*, pages 166–173, June 2016

Appendix A

Parameter estimation for SMM

A.1 Objective function

The regularized log-likelihood (objective) function is given by,

$$\mathcal{L} = \sum_{d=1}^{D} \underbrace{\sum_{i=1}^{V} x_{di} \log \theta_{di} - \frac{\lambda}{2}||\boldsymbol{w}_d||_2}_{\mathcal{L}_d} - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1 \tag{A.1}$$

$$= \sum_{d=1}^{D} \sum_{i=1}^{V} x_{di} \log \left(\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}} \right) - \frac{\lambda}{2} |||\boldsymbol{w}_d||_2 - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1, \tag{A.2}$$

where $\boldsymbol{w}_d$ is a column vector and $\boldsymbol{t}_i$ is a row vector.

It is conveniet to have derivatives of θ_{di} with respect to $\boldsymbol{w}_d$ and $\boldsymbol{t}_i$.

First, taking the derivative of θ_{di} with respect to $\boldsymbol{w}_d$:

$$\frac{\partial \theta_{di}}{\partial \boldsymbol{w}_d} = \frac{\boldsymbol{t}_i^{\mathsf{T}} \exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\} \sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} - \exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\} \sum_j \boldsymbol{t}_j^{\mathsf{T}} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}{\Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} \Big)^2} \tag{A.3}$$

$$= \frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\} \Big(\boldsymbol{t}_i^{\mathsf{T}} \sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} - \sum_j \boldsymbol{t}_j^{\mathsf{T}} \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} \Big)}{\Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} \Big) \Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} \Big)} \tag{A.4}$$

$$= \underbrace{\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{\theta_{di}} \left(\boldsymbol{t}_i^{\mathsf{T}} \underbrace{\frac{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{1} - \sum_j \boldsymbol{t}_j^{\mathsf{T}} \underbrace{\frac{\exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{\theta_{dj}} \right) \tag{A.5}$$

$$= \theta_{di} \Big(\boldsymbol{t}_i^{\mathsf{T}} - \sum_{j=1}^{V} \boldsymbol{t}_j^{\mathsf{T}} \theta_{dj} \Big) \tag{A.6}$$

$$\boxed{\frac{\partial \theta_{di}}{\partial \boldsymbol{w}_d} = \theta_{di}\Big(\boldsymbol{t}_i^{\mathsf{T}} - \sum_{j=1}^{V} \boldsymbol{t}_j^{\mathsf{T}} \theta_{dj}\Big).} \tag{A.7}$$

Next, taking the derivative of θ_{di} with respect to $\boldsymbol{t}_k$ (row in $\boldsymbol{T}$):

$$\frac{\partial \theta_{di}}{\partial \boldsymbol{t}_k} = \frac{(\delta_{ik}\boldsymbol{w}_d^{\mathsf{T}} \exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}) \sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} - \exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\} \boldsymbol{w}_d^{\mathsf{T}} \exp\{m_k + \boldsymbol{t}_k \boldsymbol{w}_d\}}{\Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}\Big)^2} \tag{A.8}$$

where δ_{ik} is the Kronecker-Delta $\delta_{ik} \triangleq \begin{cases} 1 \text{ if } \quad i = k \\ 0 \text{ otherwise.} \end{cases}$

$$= \frac{\boldsymbol{w}_d^{\mathsf{T}} \exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\} \Big(\delta_{ik} \sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\} - \exp\{m_k + \boldsymbol{t}_k \boldsymbol{w}_d\}\Big)}{\Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}\Big)\Big(\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}\Big)} \tag{A.9}$$

$$= \boldsymbol{w}_d^{\mathsf{T}} \underbrace{\frac{\exp\{m_i + \boldsymbol{t}_i \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{\theta_{di}} \Bigg(\underbrace{\delta_{ik} \frac{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{1} - \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k \boldsymbol{w}_d\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \boldsymbol{w}_d\}}}_{\theta_{dk}} \Bigg) \tag{A.10}$$

$$= \theta_{di}\Big(\delta_{ik} - \theta_{dk}\Big)\boldsymbol{w}_d^{\mathsf{T}} \tag{A.11}$$

$$\boxed{\frac{\partial \theta_{di}}{\partial \boldsymbol{t}_k} = \theta_{di}\Big(\delta_{ik} - \theta_{dk}\Big)\boldsymbol{w}_d^{\mathsf{T}}.} \tag{A.12}$$

A.1.1 Derivatives of objective

Taking the derivative of $\mathcal{L}_d$ from Eq. (A.1) with respect to $\boldsymbol{w}_d$ and using the formula from Eq. (A.7):

$$\frac{\partial \mathcal{L}_d}{\partial \boldsymbol{w}_d} = \sum_{i=1}^{V} x_{di} \left[\frac{1}{\theta_{di}} \theta_{di}\Big(\boldsymbol{t}_i^{\mathsf{T}} - \sum_j \boldsymbol{t}_j^{\mathsf{T}} \theta_{dj}\Big) \right] - \lambda \boldsymbol{w}_d \tag{A.13}$$

$$= \sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}} x_{di} - \sum_{i=1}^{V} x_{di} \sum_{j=1}^{V} \boldsymbol{t}_j^{\mathsf{T}} \theta_{dj} - \lambda \boldsymbol{w}_d, \tag{A.14}$$

interchanging indices i, j and re-arranging

$$= \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} x_{di} - \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \theta_{di} \sum_{j=1}^{V} x_{dj} - \lambda \boldsymbol{w}_d \tag{A.15}$$

$$= \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \left[x_{di} - \theta_{di} \sum_{k=1}^{V} x_{dj} \right] - \lambda \boldsymbol{w}_d. \tag{A.16}$$

$$\boxed{\nabla_{\boldsymbol{w}_d} \mathcal{L} = \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \left[x_{di} - \theta_{di} \sum_{j=1}^{V} x_{dj} \right] - \lambda \boldsymbol{w}_d.} \tag{A.17}$$

Taking the derivative of Eq. (A.17) with respect to $\boldsymbol{w}_d$ and making use of Eq. (A.7):

$$\frac{\partial^2 \mathcal{L}_d}{\partial \boldsymbol{w}_d \partial \boldsymbol{w}_d^\mathsf{T}} = \frac{\partial}{\partial \boldsymbol{w}_d^\mathsf{T}} \left[\sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \left[x_{di} - \theta_{di} \sum_{k=1}^{V} x_{dk} \right] - \lambda \boldsymbol{w}_d \right] \tag{A.18}$$

$$= \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \left[0 - \theta_{di} (\boldsymbol{t}_i - \sum_{j} \boldsymbol{t}_j \theta_{dj}) \sum_{k=1}^{V} x_{dk} \right] - \lambda \boldsymbol{I} \tag{A.19}$$

$$\boxed{\nabla^2_{\boldsymbol{w}_d} \mathcal{L}_d = \boldsymbol{H}_{\boldsymbol{w}_d}(\mathcal{L}_d) = - \sum_{i=1}^{V} \boldsymbol{t}_i^\mathsf{T} \theta_{di} \Big(\boldsymbol{t}_i - \sum_{j=1}^{V} \boldsymbol{t}_j \theta_{dj} \Big) \sum_{k=1}^{V} x_{dk}} \tag{A.20}$$

Derivatives of objective with respect to $\boldsymbol{t}_k$

Taking the derivative of $\mathcal{L}$ from Eq. (A.1) with respect to $\boldsymbol{t}_k$ and using the formula from Eq. (A.12):

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{t}_k} = \left[\sum_{d=1}^{D} \Big(\sum_{i=1}^{V} x_{di} \frac{1}{\theta_{di}} \theta_{di} (\delta_{ik} - \theta_{dk}) \Big) \boldsymbol{w}_d^\mathsf{T} \right] - \omega \operatorname{sign}(\boldsymbol{t}_k) \tag{A.21}$$

$$= \left[\sum_{d=1}^{D} \Big(\sum_{i=1}^{V} x_{di} \delta_{ik} - \sum_{i=1}^{V} x_{di} \theta_{dk} \Big) \boldsymbol{w}_d^\mathsf{T} \right] - \omega \operatorname{sign}(\boldsymbol{t}_k) \tag{A.22}$$

$$= \left[\sum_{d=1}^{D} \Big(x_{dk} - \theta_{dk} \sum_{i=1}^{V} x_{di} \Big) \boldsymbol{w}_d^\mathsf{T} \right] - \omega \operatorname{sign}(\boldsymbol{t}_k) \tag{A.23}$$

$$\boxed{\nabla_{\boldsymbol{t}_k} \mathcal{L} = \left[\sum_{d=1}^{D} \Big(x_{dk} - \theta_{dk} \sum_{i=1}^{V} x_{di} \Big) \boldsymbol{w}_d^\mathsf{T} \right] - \omega \operatorname{sign}(\boldsymbol{t}_k)} \tag{A.24}$$

Here sign operates element-wise.

Taking the derivative of Eq. (A.24) with respect to $\boldsymbol{t}_l$ and making use of Eq. (A.12):

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{t}_k \partial \boldsymbol{t}_l^\mathsf{T}} = \frac{\partial}{\partial \boldsymbol{t}_l^\mathsf{T}} \left[\Big(\sum_{d=1}^{D} x_{dk} \Big) - \Big(\sum_{d=1}^{D} \theta_{dk} \sum_{i=1}^{V} x_{di} \boldsymbol{w}_d^\mathsf{T} \Big) - \omega \, \mathrm{sign}(\boldsymbol{t}_k) \right] \tag{A.25}$$

$$= - \sum_{d=1}^{D} \Big(\theta_{dk} (\delta_{kl} - \theta_{dl}) \sum_{i=1}^{V} x_{di} \Big) \boldsymbol{w}_d \boldsymbol{w}_d^\mathsf{T} \tag{A.26}$$

$$\boxed{\nabla_{\boldsymbol{t}_l} (\nabla_{\boldsymbol{t}_k} \mathcal{L}) = \boldsymbol{H}_{\boldsymbol{t}_k \boldsymbol{t}_l} (\mathcal{L}) = - \sum_{d=1}^{D} \Big(\theta_{dk} (\delta_{kl} - \theta_{dl}) \sum_{i=1}^{V} x_{di} \Big) \boldsymbol{w}_d \boldsymbol{w}_d^\mathsf{T}.} \tag{A.27}$$

Appendix B

Variational Bayes for Bayesian SMM

Let $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{\nu}, \boldsymbol{\Gamma}^{-1})$ denote the variational distribution, and $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ denote prior over $\boldsymbol{w}$. The following derivations are made assuming full precision matrices for the Gaussian distributions. However, the results will be provided for the isotropic and diagonal covariances as described in Chapter 5.

B.1 Variational lower-bound (ELBO)

The variational lower bound (or evidence lower bound, ELBO) for a document[1] is given as follows:

$$\mathcal{L}(q) = -\underbrace{D_{\mathrm{KL}}(q \,||\, p_0)}_{\mathsf{A}} + \underbrace{\mathbb{E}_q[\log p(\boldsymbol{x} \,|\, \boldsymbol{w})]}_{\mathsf{B}}. \tag{B.1}$$

Term A in the above equation is the KL divergence from variational distribution to the prior:

$$D_{\mathrm{KL}}(q \,||\, p_0) = -\int q(\boldsymbol{w}) \log\left(\frac{p(\boldsymbol{w})}{q(\boldsymbol{w})}\right) \mathrm{d}\boldsymbol{w} \tag{B.2}$$

$$= -\Big[\underbrace{\int q(\boldsymbol{w}) \log p(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}_{\mathbb{E}_q[\log p(\boldsymbol{w})]} \underbrace{- \int q(\boldsymbol{w}) \log q(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}_{\mathrm{H}[q]}\Big]. \tag{B.3}$$

Solving $\mathrm{H}[q]$ (entropy of variational distribution):

$$\mathrm{H}[q] = -\int q(\boldsymbol{w}) \log q(\boldsymbol{w}) \, \mathrm{d}\boldsymbol{w}$$

$$= -\int q(\boldsymbol{w}) \Big[\log\left(\frac{|\boldsymbol{\Gamma}|^{1/2}}{(2\pi)^{K/2}}\right) - \frac{1}{2}(\boldsymbol{w}-\boldsymbol{\nu})^{\mathsf{T}} \boldsymbol{\Gamma} (\boldsymbol{w}-\boldsymbol{\nu})\Big] \, \mathrm{d}\boldsymbol{w} \tag{B.4}$$

$$= \frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| + \frac{1}{2}\int q(\boldsymbol{w})[(\boldsymbol{w}-\boldsymbol{\nu})^{\mathsf{T}} \boldsymbol{\Gamma} (\boldsymbol{w}-\boldsymbol{\nu})] \, \mathrm{d}\boldsymbol{w} \tag{B.5}$$

$$= \frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| + \frac{1}{2}\underbrace{\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\nu})^{\mathsf{T}} \boldsymbol{\Gamma} (\boldsymbol{w}-\boldsymbol{\nu})]}_{\mathsf{A}_1} \tag{B.6}$$

[1] Document suffix d is ignored for brevity.

Solving A_1:

$$\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\nu})^\top \boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\nu})] = \mathbb{E}_q\Big[\mathrm{Tr}((\boldsymbol{w}-\boldsymbol{\nu})^\top \boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\nu}))\Big] \tag{B.7}$$

$$= \mathbb{E}_q\Big[\mathrm{Tr}(\boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\nu}))(\boldsymbol{w}-\boldsymbol{\nu})^\top\Big] \tag{B.8}$$

$$= \mathrm{Tr}\Big(\mathbb{E}_q[\boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\nu})(\boldsymbol{w}-\boldsymbol{\nu})^\top]\Big) \tag{B.9}$$

$$= \mathrm{Tr}\Big(\boldsymbol{\Gamma}\,\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\nu})(\boldsymbol{w}-\boldsymbol{\nu})^\top]\Big) \tag{B.10}$$

$$= \mathrm{Tr}(\boldsymbol{\Gamma}\,\boldsymbol{\Gamma}^{-1}) \tag{B.11}$$

$$= K. \tag{B.12}$$

From Eqs. (B.6) and (B.12), we have:

$$\boxed{\mathrm{H}[q] = \frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| + \frac{K}{2}.} \tag{B.13}$$

Solving $\mathbb{E}_q[\log p(\boldsymbol{w})]$ from Eq. (B.3) (expectation of log-Gaussian with respect to the Gaussian):

$$\mathbb{E}_q[\log p(\boldsymbol{w})] = \mathbb{E}_q\Big[\log\left(\frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{K/2}}\right) - \frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{w}-\boldsymbol{\mu})\Big]$$

$$= -\frac{K}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\underbrace{\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{w}-\boldsymbol{\mu})]}_{\mathsf{A}_2}. \tag{B.14}$$

Solving A_2:

$$\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{w}-\boldsymbol{\mu})] = \mathbb{E}_q\Big[\mathrm{Tr}((\boldsymbol{w}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{w}-\boldsymbol{\mu}))\Big] \tag{B.15}$$

$$= \mathrm{Tr}\Big(\boldsymbol{\Lambda}\,\mathbb{E}_q[(\boldsymbol{w}-\boldsymbol{\mu})(\boldsymbol{w}-\boldsymbol{\mu})^\top]\Big) \tag{B.16}$$

$$= \mathrm{Tr}\Big(\boldsymbol{\Lambda}\,(\,\mathbb{E}_q[\boldsymbol{w}\boldsymbol{w}^\top] - \mathbb{E}_q[\boldsymbol{w}]\boldsymbol{\mu}^\top - \boldsymbol{\mu}\mathbb{E}_q[\boldsymbol{w}^\top] + \mathbb{E}_q[\boldsymbol{\mu}\boldsymbol{\mu}^\top]\,)\Big) \tag{B.17}$$

$$= \mathrm{Tr}\Big(\boldsymbol{\Lambda}\,(\,\boldsymbol{\Gamma}^{-1} + \boldsymbol{\nu}\boldsymbol{\nu}^\top - \boldsymbol{\nu}\boldsymbol{\mu}^\top - \boldsymbol{\mu}\boldsymbol{\nu}^\top + \boldsymbol{\mu}\boldsymbol{\mu}^\top\,)\Big) \tag{B.18}$$

$$= \mathrm{Tr}\Big(\boldsymbol{\Lambda}\,(\,\boldsymbol{\Gamma}^{-1} + (\boldsymbol{\nu}-\boldsymbol{\mu})(\boldsymbol{\nu}-\boldsymbol{\mu})^\top\,)\Big) \tag{B.19}$$

$$= \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{-1}) + \mathrm{Tr}(\boldsymbol{\Lambda}(\boldsymbol{\nu}-\boldsymbol{\mu})(\boldsymbol{\nu}-\boldsymbol{\mu})^\top) \tag{B.20}$$

$$= \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{-1}) + (\boldsymbol{\nu}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\nu}-\boldsymbol{\mu}). \tag{B.21}$$

From Eqs. (B.14) and (B.21), we have:

$$\boxed{\mathbb{E}_q[\log p(\boldsymbol{w})] = -\frac{K}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{-1}) - \frac{1}{2}(\boldsymbol{\nu}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\nu}-\boldsymbol{\mu}).} \tag{B.22}$$

Combining Eqs. (B.13) and (B.22), we get the KL divergence from variational distribution (multivariate Gaussian) to the prior (multivariate Gaussian):

$$\boxed{D_{\mathrm{KL}}(q||p) = \frac{1}{2}\Big[\mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{-1}) + \log|\boldsymbol{\Gamma}| - \log|\boldsymbol{\Lambda}| + (\boldsymbol{\nu}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\nu}-\boldsymbol{\mu}) - K\Big].} \tag{B.23}$$

Term B from Eq. (B.1) is the expectation of log-likelihood of the data with respect to the variational distribution $q(\boldsymbol{w})$:

$$\mathbb{E}_q[\log p(\boldsymbol{x} \mid \boldsymbol{w})] = \mathbb{E}_q\Big[\sum_{i=1}^{V} x_i \log\Big(\frac{\exp\{m_i + \boldsymbol{t}_i\boldsymbol{w}\}}{\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\boldsymbol{w}\}}\Big)\Big] \tag{B.24}$$

$$= \sum_{i=1}^{V} x_i \left[\mathbb{E}_q[m_i + \boldsymbol{t}_i\boldsymbol{w}] - \mathbb{E}_q[\log\left(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\boldsymbol{w}\}\right)]\right] \tag{B.25}$$

$$= \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}) - \underbrace{\mathbb{E}_q[\log\left(\sum_{j=1}^{V}\exp\{m_j + \boldsymbol{t}_j\boldsymbol{w}\}\right)]}_{\mathcal{F}(\boldsymbol{w})}\right]. \tag{B.26}$$

$\mathcal{F}(\boldsymbol{w})$ is the expectation over log-sum-exp, which is intractable. The following two approaches are used to approximate $\mathcal{F}(\boldsymbol{w})$:

1. Jensen's inequality

2. Monte-Carlo approximation via re-parametrization trick.

B.1.1 ELBO with Jensen's inequality (ELBO_{JI})

Applying Jensen's inequality on $\mathcal{F}(\boldsymbol{w})$:

$$\mathbb{E}_q[\log\left(\sum_{j=1}^{V}\exp(m_j + \boldsymbol{t}_j\boldsymbol{w})\right)] \leq \log\left(\sum_{j=1}^{V}\mathbb{E}_q[\exp(m_j + \boldsymbol{t}_j\boldsymbol{w})]\right)$$

$$= \log\left(\sum_{j=1}^{V}\exp(m_j)\,\mathbb{E}_q[\exp(\boldsymbol{t}_j\boldsymbol{w})]\right). \tag{B.27}$$

Because of this inequality, we can only have a lower bound on the expectation from Eq. (B.26),

$$\mathbb{E}_q[\log p(\boldsymbol{x} \mid \boldsymbol{w})] \geq \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i\boldsymbol{\nu}) - \log\left(\sum_{j=1}^{V}\exp(m_j)\,\underbrace{\mathbb{E}_q[\exp(\boldsymbol{t}_j\boldsymbol{w})]}_{\mathsf{A}_2}\right)\right] \tag{B.28}$$

Solving A_2:

$$
\begin{aligned}
\mathbb{E}_q[\exp(\boldsymbol{t}_j\boldsymbol{w})] &= \int q(\boldsymbol{w})\exp(\boldsymbol{t}_j\boldsymbol{w})\,\mathrm{d}\boldsymbol{w} \\
&= \int \underbrace{\frac{|\boldsymbol{\Gamma}|^{1/2}}{(2\pi)^{K/2}}}_{k_1}\exp\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\nu})^\top\boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\nu})+\boldsymbol{t}_j\boldsymbol{w}\}\,\mathrm{d}\boldsymbol{w} \\
&= \int k_1\,\exp\Big\{-\frac{1}{2}\Big(\boldsymbol{w}^\top\boldsymbol{\Gamma}\boldsymbol{w}-2\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{w}-2\boldsymbol{t}_j\boldsymbol{w}+\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}\Big)\Big\}\,\mathrm{d}\boldsymbol{w} \\
&= \int k_1\,\exp\Big\{-\frac{1}{2}\Big(\boldsymbol{w}^\top\boldsymbol{\Gamma}\boldsymbol{w}-2\underbrace{(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}+\boldsymbol{t}_j)}_{\boldsymbol{c}^\top}\boldsymbol{w}\Big)\Big\}\underbrace{\exp\Big\{-\frac{1}{2}\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}\Big\}}_{k_2}\,\mathrm{d}\boldsymbol{w} \\
&= k_2\int k_1\,\exp\Big\{-\frac{1}{2}\Big(\boldsymbol{w}^\top\boldsymbol{\Gamma}\boldsymbol{w}-2\boldsymbol{c}^\top\boldsymbol{w}\Big)\Big\}\,\mathrm{d}\boldsymbol{w} \\
&= k_2\int k_1\,\exp\Big\{-\frac{1}{2}\Big(\underbrace{\boldsymbol{w}^\top\boldsymbol{\Gamma}\boldsymbol{w}-2\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\boldsymbol{w}+\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{-1}\boldsymbol{c}}_{\text{completion of square}}-\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{-1}\boldsymbol{c}\Big)\Big\}\,\mathrm{d}\boldsymbol{w} \\
&= k_2\int k_1\,\exp\Big\{-\frac{1}{2}((\boldsymbol{w}-\boldsymbol{\Gamma}^{-1}\boldsymbol{c})^\top\boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\Gamma}^{-1}\boldsymbol{c}))\Big\}\exp\Big\{\frac{1}{2}\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{c}\Big\}\,\mathrm{d}\boldsymbol{w} \\
&= k_2\exp\Big\{\frac{1}{2}\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{c}\Big\}\underbrace{\int\frac{1}{(2\pi)^{K/2}|\boldsymbol{\Gamma}^{-1}|^{1/2}}\exp\Big\{-\frac{1}{2}((\boldsymbol{w}-\boldsymbol{\Gamma}^{-1}\boldsymbol{c})^\top\boldsymbol{\Gamma}(\boldsymbol{w}-\boldsymbol{\Gamma}^{-1}\boldsymbol{c}))\Big\}\,\mathrm{d}\boldsymbol{w}}_{1} \\
&= \exp\Big\{-\frac{1}{2}(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}-\boldsymbol{c}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{c})\Big\} \\
&= \exp\Big\{-\frac{1}{2}(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}-(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}+\boldsymbol{t}_j)\boldsymbol{\Gamma}^{-1}(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}+\boldsymbol{t}_j)^\top)\Big\} \\
&= \exp\Big\{-\frac{1}{2}(\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}-\boldsymbol{\nu}^\top\boldsymbol{\Gamma}\boldsymbol{\nu}-\boldsymbol{\nu}^\top\boldsymbol{t}_j^\top-\boldsymbol{t}_j\boldsymbol{\nu}-\boldsymbol{t}_j\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_j^\top)\Big\} \\
&= \exp\{\boldsymbol{t}_j\boldsymbol{\nu}+\frac{1}{2}\boldsymbol{t}_j\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_j^\top\}. \qquad \text{(B.29)}
\end{aligned}
$$

From Eqs.(B.28), and (B.29), we have:

$$
\mathbb{E}_q[\log p(\boldsymbol{x}\mid\boldsymbol{w})] \geq \sum_{i=1}^{V} x_i\left[(m_i+\boldsymbol{t}_i\boldsymbol{\nu})-\log\left(\sum_{j=1}^{V}\exp\left(m_j+\boldsymbol{t}_j\boldsymbol{\nu}+\frac{1}{2}\boldsymbol{t}_j\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_j^\top\right)\right)\right]. \qquad \text{(B.30)}
$$

Combining Eqs.(B.23) and (B.30), gives the *lower-bound* on complete evidence lower-bound (ELBO), $\mathcal{L}_{\text{JI}}(q_d)$ per each document d:

$$
\begin{aligned}
\mathcal{L}_{\text{JI}}(q_d) \geq \sum_{i=1}^{V} x_i &\left[(m_i+\boldsymbol{t}_i\boldsymbol{\nu}_d)-\log\left(\sum_{j=1}^{V}\exp\left(m_j+\boldsymbol{t}_j\boldsymbol{\nu}_d+\frac{1}{2}\boldsymbol{t}_j\boldsymbol{\Gamma}_d^{-1}\boldsymbol{t}_j^\top\right)\right)\right] \\
&-\frac{1}{2}\left[K+\log|\boldsymbol{\Lambda}|-\log|\boldsymbol{\Gamma}_d|-\mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{\Gamma}_d^{-1})-(\boldsymbol{\nu}_d-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}(\boldsymbol{\nu}_d-\boldsymbol{\mu})\right]. \qquad \text{(B.31)}
\end{aligned}
$$

If the prior distribution is $p(w) = \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, (\lambda \boldsymbol{I})^{-1})$, and the variational posterior is diagonal Gaussian with the following parametrization:

$$q(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{\nu}_d, \text{diag}(\exp\{2\boldsymbol{\varsigma}_d\})), \tag{B.32}$$

then ELBO from Eq. (B.31) becomes:

$$\boxed{\begin{aligned}\mathcal{L}_{\text{JI}}(q_d) \geq \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}_d) - \log \left(\sum_{j=1}^{V} \exp\left(m_j + \boldsymbol{t}_j \boldsymbol{\nu}_d + \frac{1}{2} \boldsymbol{t}_j \, \text{diag}(\exp\{2\boldsymbol{\varsigma}_d\}) \, \boldsymbol{t}_j^{\mathsf{T}} \right) \right) \right] \\ -\frac{1}{2} \left[\lambda \, \text{Tr}(\text{diag}(\exp\{2\boldsymbol{\varsigma}_d\})) - \log|\text{diag}(\exp\{2\boldsymbol{\varsigma}_d\})| - K \, \log \lambda + \lambda \boldsymbol{\nu}_d^{\mathsf{T}} \boldsymbol{\nu}_d - K \right].\end{aligned}} \tag{B.33}$$

The complete objective is the summation of $\mathcal{L}_{\text{JI}}(q_d)$ for all the documents along with the regularization term for the rows in matrix $\boldsymbol{T}$:

$$\boxed{\text{ELBO}_{\text{JI}} = \mathcal{L}_{\text{JI}} \geq \sum_{d=1}^{D} \mathcal{L}_{\text{JI}}(q_d) - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1.} \tag{B.34}$$

B.1.2 ELBO with Re-parametrization trick (ELBO_{RP})

The empirical approximation of $\mathcal{F}(\boldsymbol{w})$ from Eq. (B.26) is given by:

$$\mathbb{E}_{q(\boldsymbol{w})}[\mathcal{F}(\boldsymbol{w})] \approx \frac{1}{R} \sum_{r=1}^{R} \mathcal{F}(\tilde{\boldsymbol{w}}_r) \tag{B.35}$$

where $\tilde{\boldsymbol{w}}_r \, \forall \, r = 1 \ldots R$, represents samples drawn from $q(\boldsymbol{w})$, and $\mathcal{F}(\tilde{\boldsymbol{w}}_r)$ implies the function evaluated at $\tilde{\boldsymbol{w}}_r$. We will re-parametrize the random variable $\boldsymbol{w}$ using a differentiable transformation function $g(\boldsymbol{\epsilon})$ over another random (auxiliary) variable $\boldsymbol{\epsilon}$. This will allows us to express the random variable $\boldsymbol{w}$ as deterministic, i.e., $\boldsymbol{w} = g(\boldsymbol{\epsilon})$. Using this re-parametrization trick, the empirical approximation of $\mathcal{F}$ is given as follows:

$$\mathbb{E}_{q(\boldsymbol{w})}[\mathcal{F}(\boldsymbol{w})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}[\mathcal{F}\left(g(\boldsymbol{\epsilon})\right)] \tag{B.36}$$

$$\approx \frac{1}{R} \sum_{r=1}^{R} \mathcal{F}\left(g(\tilde{\boldsymbol{\epsilon}}_r)\right), \tag{B.37}$$

where $\tilde{\boldsymbol{\epsilon}}_r \, \forall \, r = 1 \ldots R$ represents samples drawn from $p(\boldsymbol{\epsilon})$, and $g(\tilde{\boldsymbol{\epsilon}}_r)$ implies the function evaluated at $\tilde{\boldsymbol{\epsilon}}_r$. If $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \boldsymbol{I})$, then $\boldsymbol{w} = g(\boldsymbol{\epsilon}) = \boldsymbol{\nu} + \boldsymbol{L}\,\boldsymbol{\epsilon}$, where, $\boldsymbol{L}\boldsymbol{L}^{\mathsf{T}} = \boldsymbol{\Gamma}^{-1}$ (Cholesky decomposition).

The empirical approximation to $\mathcal{F}(\boldsymbol{w})$ is given by:

$$\mathcal{F}(\boldsymbol{w}) \approx \frac{1}{R} \sum_{r=1}^{R} \log \left(\sum_{j}^{V} \exp\{m_j + \boldsymbol{t}_j \, g(\boldsymbol{\epsilon}_r)\} \right). \tag{B.38}$$

From Eqs. (B.26) and (B.38), we have:

$$\mathbb{E}_q[\log p(\boldsymbol{x} \mid \boldsymbol{w})] \approx \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}) - \frac{1}{R} \sum_{r=1}^{R} \log \left(\sum_{j}^{V} \exp(m_j + \boldsymbol{t}_j \, g(\, \boldsymbol{\epsilon}_r \,)) \right) \right]. \tag{B.39}$$

Combining Eqs. (B.23) and (B.39), gives the *approximation* on evidence lower-bound (ELBO), $\mathcal{L}_{\mathsf{RP}}(q_d)$ for each document d:

$$\begin{aligned} \mathcal{L}_{\mathsf{RP}}(q_d) \approx \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}_d) - \frac{1}{R} \sum_{r=1}^{R} \log \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \, g(\boldsymbol{\epsilon}_{dr})\} \right) \right] \\ - \frac{1}{2} \left[K + \log|\boldsymbol{\Lambda}| - \log|\boldsymbol{\Gamma}_d| - \mathrm{Tr}(\boldsymbol{\Lambda} \boldsymbol{\Gamma}_d^{-1}) - (\boldsymbol{\nu}_d - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Lambda} (\boldsymbol{\nu}_d - \boldsymbol{\mu}) \right] \end{aligned} \tag{B.40}$$

If the prior distribution is $p(w) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, (\lambda \boldsymbol{I})^{-1})$, and the variational posterior is diagonal Gaussian with the following parametrization:

$$q(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{\nu}_d, \mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})), \tag{B.41}$$

then ELBO from Eq. (B.40) becomes:

$$\boxed{\begin{aligned} \mathcal{L}_{\mathsf{RP}}(q_d) \approx \sum_{i=1}^{V} x_i \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}_d) - \frac{1}{R} \sum_{r=1}^{R} \log \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \, g(\boldsymbol{\epsilon}_{dr})\} \right) \right] \\ - \frac{1}{2} \left[\lambda \, \mathrm{Tr}(\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})) - \log|\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})| - K \, \log \lambda + \lambda \boldsymbol{\nu}_d^{\mathsf{T}} \boldsymbol{\nu}_d - K \right]. \end{aligned}} \tag{B.42}$$

The complete objective is the summation of $\mathcal{L}_{\mathsf{RP}}(q_d)$ for all the documents along with the regularization term for the rows in matrix $\boldsymbol{T}$:

$$\boxed{\mathrm{ELBO}_{\mathsf{RP}} = \mathcal{L}_{\mathsf{RP}} \approx \sum_{d=1}^{D} \mathcal{L}_{\mathsf{RP}}(q_d) - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1.} \tag{B.43}$$

B.2 Inference in Bayesian SMM

During inference, we restricted the variational distribution to be diagonal Gaussian with the following parametrization:

$$q(\boldsymbol{w}_d) = \mathcal{N}(\boldsymbol{w}_d \mid \boldsymbol{\nu}_d, \mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})). \tag{B.44}$$

It is convenient to have the derivative of the KL divergence term with respect to the variational parameters.

$$D_{\mathrm{KL}}(q||p) = \frac{1}{2}\Big[\lambda \,\mathrm{Tr}(\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})) - \log|\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})| - K\,\log\lambda + \lambda\boldsymbol{\nu}_d^{\mathsf{T}}\boldsymbol{\nu}_d - K\Big] \tag{B.45}$$

$$\frac{\partial D_{\mathrm{KL}}(q||p)}{\partial \boldsymbol{\nu}_d} = \lambda\boldsymbol{\nu}_d \tag{B.46}$$

$$\frac{\partial D_{\mathrm{KL}}(q||p)}{\partial \boldsymbol{\varsigma}_d} = \lambda\exp\{2\boldsymbol{\varsigma}_d\} - \mathbf{1} \tag{B.47}$$

B.2.1 Gradients of ELBO$_{\text{JI}}$

VB E-step: Updating the parameters of variational distribution:

Taking derivative of the objective function Eq. (B.33) with respect to mean parameter $\boldsymbol{\nu}_d$ of variational distribution $q(\boldsymbol{w}_d)$ specific to a single document d, and using Eq. (B.46):

$$\frac{\partial \mathcal{L}_{\text{JI}}(q_d)}{\partial \boldsymbol{\nu}_d} = \sum_{i=1}^{V} x_{di}\Big[\boldsymbol{t}_i^{\mathsf{T}} - \sum_k \boldsymbol{t}_k^{\mathsf{T}} \underbrace{\frac{\exp\Big(m_k + \boldsymbol{t}_k\boldsymbol{\nu}_d + \frac{1}{2}\boldsymbol{t}_k\,\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_k^{\mathsf{T}}\Big)}{\sum_j \exp\Big(m_j + \boldsymbol{t}_j\boldsymbol{\nu}_d + \frac{1}{2}\boldsymbol{t}_j\,\mathrm{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_j^{\mathsf{T}}\Big)}}_{\varphi_{dk}}\Big] - \lambda\boldsymbol{\nu}_d \tag{B.48}$$

$$\begin{aligned}
&= \sum_i x_{di}\Big[\boldsymbol{t}_i^{\mathsf{T}} - \sum_k \boldsymbol{t}_k^{\mathsf{T}}\varphi_{dk}\Big] - \lambda\boldsymbol{\nu}_d \\
&= \Big[\sum_i x_{di}\boldsymbol{t}_i^{\mathsf{T}} - \sum_i x_{di}\sum_k \boldsymbol{t}_k^{\mathsf{T}}\varphi_{dk}\Big] - \lambda\boldsymbol{\nu}_d \\
&= \Big[\sum_i x_{di}\boldsymbol{t}_i^{\mathsf{T}} - \sum_i \boldsymbol{t}_i^{\mathsf{T}}\varphi_{di}\sum_k x_{dk}\Big] - \lambda\boldsymbol{\nu}_d
\end{aligned}$$

$$\boxed{\nabla_{\boldsymbol{\nu}_d}\mathcal{L}_{\text{JI}}(q_d) = \Big[\sum_{i=1}^{V} \boldsymbol{t}_i^{\mathsf{T}}(x_{di} - \varphi_{di}\sum_{k=1}^{V} x_{dk})\Big] - \lambda\boldsymbol{\nu}_d.} \tag{B.49}$$

Taking derivative of the objective function Eq. (B.33) with respect to log standard-deviation parameter $\boldsymbol{\varsigma}$ of variational distribution $q(\boldsymbol{w}_d)$ specific to a single document d, and using Eq. (B.47):

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{JI}}(q_d)}{\partial \boldsymbol{\varsigma}_d} &= \sum_{i=1}^{V} x_{di} \Big[-\sum_k \frac{1}{2}\boldsymbol{t}_k^{\mathsf{T}} \odot 2\exp\{2\boldsymbol{\varsigma}\} \odot \boldsymbol{t}_k^{\mathsf{T}} \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k\boldsymbol{\nu} + \frac{1}{2}\boldsymbol{t}_k \operatorname{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_k^{\mathsf{T}}\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j\boldsymbol{\nu} + \frac{1}{2}\boldsymbol{t}_j \operatorname{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_j^{\mathsf{T}}\}}}_{\varphi_{dk}} \Big] \\ &\quad - \lambda \exp\{2\boldsymbol{\varsigma}_d\} + \mathbf{1} \\ &= -\sum_{i=1}^{V} x_{di} \sum_{k=1}^{V} \boldsymbol{t}_k^{\mathsf{T}} \odot \exp\{2\boldsymbol{\varsigma}\} \odot \boldsymbol{t}_k^{\mathsf{T}}\, \varphi_{dk} - \lambda \exp\{2\boldsymbol{\varsigma}_d\} + \mathbf{1}\end{aligned}$$

$$\boxed{\nabla_{\boldsymbol{\varsigma}_d} \mathcal{L}_{\text{JI}}(q_d) = \mathbf{1} - \lambda \exp\{2\boldsymbol{\varsigma}_d\} - \sum_{i=1}^{V} x_{di} \sum_{k=1}^{V} \boldsymbol{t}_k^{\mathsf{T}} \odot \exp\{2\boldsymbol{\varsigma}\} \odot \boldsymbol{t}_k^{\mathsf{T}}\, \varphi_{dk}.} \tag{B.50}$$

VB M-step: Updating model parameters:

Taking derivative of the objective function Eq. (B.34) with respect to a row $\boldsymbol{t}_k$ in matrix $\boldsymbol{T}$:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{JI}}}{\partial \boldsymbol{t}_k} &= \frac{\partial}{\partial \boldsymbol{t}_k} \sum_{d=1}^{D}\sum_{i=1}^{V} x_{di} \Bigg[(m_i + \boldsymbol{t}_i^{\mathsf{T}}\boldsymbol{\nu}_d) \\ &\qquad - \log\Bigg(\sum_{j=1}^{V} \exp\{m_i + \boldsymbol{t}_j\boldsymbol{\nu}_d + \frac{1}{2}\boldsymbol{t}_j \operatorname{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_j^{\mathsf{T}}\} \Bigg) \Bigg] - \lambda \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1 \\ &= \sum_{d=1}^{D} \Bigg[x_{dk}\boldsymbol{\nu}_d^{\mathsf{T}} - \sum_{i=1}^{V} x_{di} \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k\boldsymbol{\nu} + \frac{1}{2}\boldsymbol{t}_k \operatorname{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_k^{\mathsf{T}}\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j\boldsymbol{\nu} + \frac{1}{2}\boldsymbol{t}_j \operatorname{diag}(\exp\{2\boldsymbol{\varsigma}_d\})\,\boldsymbol{t}_j^{\mathsf{T}}\}}}_{\varphi_{dk}} \Big(\boldsymbol{\nu}_d^{\mathsf{T}} + \boldsymbol{t}_k \odot \exp\{2\boldsymbol{\varsigma}_d\} \Big) \Bigg] \\ &\quad - \lambda \operatorname{sign}(\boldsymbol{t}_k) \\ &= \sum_{d=1}^{D} \Bigg[x_{dk}\boldsymbol{\nu}_d^{\mathsf{T}} - \varphi_{dk} \Big(\boldsymbol{\nu}_d^{\mathsf{T}} + \boldsymbol{t}_k \odot \exp\{2\boldsymbol{\varsigma}_d\} \Big) \sum_{i=1}^{V} x_{di} \Bigg] - \lambda \operatorname{sign}(\boldsymbol{t}_k)\end{aligned}$$

$$\boxed{\boldsymbol{\nabla}_{\boldsymbol{t}_k} \mathcal{L}_{\text{JI}} = \sum_{d=1}^{D} \Bigg[\Big(x_{dk} - \varphi_{dk} \sum_{i=1}^{V} x_{di} \Big) \boldsymbol{\nu}_d^{\mathsf{T}} - \Big(\varphi_{dk}\boldsymbol{t}_k \odot \exp\{2\boldsymbol{\varsigma}_d\} \Big) \sum_{i=1}^{V} x_{di} \Bigg] - \lambda \operatorname{sign}(\boldsymbol{t}_k).} \tag{B.51}$$

Here, sign operates element-wise and $\odot$ indicates element wise product.

B.2.2 Gradients of ELBO$_{\text{RP}}$

$$g(\boldsymbol{\epsilon}) = \boldsymbol{\nu} + \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}}, \tag{B.52}$$

where $\tilde{\boldsymbol{\epsilon}}$ represents a sample drawn from $\mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \boldsymbol{I})$.
It is convenient to have the following derivatives:

$$\frac{\partial(\boldsymbol{\nu} + \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}})}{\partial \boldsymbol{\nu}} = \boldsymbol{I} \tag{B.53}$$

$$\frac{\partial(\boldsymbol{\nu} + \exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}})}{\partial \boldsymbol{\varsigma}} = \text{diag}(\exp\{\boldsymbol{\varsigma}\} \odot \tilde{\boldsymbol{\epsilon}}) \tag{B.54}$$

VB E-step: Updating the parameters of variational distribution:

Taking derivative of the objective function Eq. (B.42) with respect to mean parameter $\boldsymbol{\nu}$ and using Eqs. (B.46), (B.53):

$$\frac{\partial \mathcal{L}_{\text{RP}}(q_d)}{\partial \boldsymbol{\nu}_d} = \sum_{i=1}^{V} x_{di} \Bigg[\boldsymbol{t}_i^\intercal - \frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} \boldsymbol{t}_k^\intercal \boldsymbol{I} \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k\, g(\boldsymbol{\epsilon}_{dr})\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j\, g(\boldsymbol{\epsilon}_{dr})\}}}_{\theta_{dkr}} \Bigg] - \lambda \boldsymbol{\nu}_d \tag{B.55}$$

$$= \Big[\sum_{i=1}^{V} x_{di} \boldsymbol{t}_i^\intercal - \sum_{i=1}^{V} \boldsymbol{t}_i^\intercal \frac{1}{R} \sum_{r=1}^{R} \theta_{dir} \sum_{k=1}^{V} x_{dk} \Big] - \lambda \boldsymbol{\nu}_d \tag{B.56}$$

$$= \Big[\sum_{i=1}^{V} \boldsymbol{t}_i^\intercal (x_{di} - \frac{1}{R} \sum_{r=1}^{R} \theta_{dir} \sum_{k=1}^{V} x_{dk}) \Big] - \lambda \boldsymbol{\nu}_d \tag{B.57}$$

$$\boxed{\nabla_{\boldsymbol{\nu}_d} \mathcal{L}_{\text{RP}}(q_d) = \Big[\sum_{i=1}^{V} \boldsymbol{t}_i^\intercal (x_i - \frac{1}{R} \sum_{r=1}^{R} \theta_{ir} \sum_{k=1}^{V} x_k) \Big] - \lambda \boldsymbol{\nu}_d} \tag{B.58}$$

Taking the derivative of objective function Eq. (B.42) with respect to $\boldsymbol{\varsigma}$ and using Eqs. (B.47), (B.54):

$$\frac{\partial \mathcal{L}_{\text{RP}}(q_d)}{\partial \boldsymbol{\varsigma}_d} = \sum_{i=1}^{V} x_{di} \Bigg[- \frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} \boldsymbol{t}_k^\intercal \text{diag}(\exp\{\boldsymbol{\varsigma}_d\} \odot \tilde{\boldsymbol{\epsilon}}_{dr}) \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k g(\boldsymbol{\epsilon}_{dr})\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j g(\boldsymbol{\epsilon}_r)\}}}_{\theta_{dkr}} \Bigg]$$

$$- \lambda \exp\{2\boldsymbol{\varsigma}_d\} + \mathbf{1} \tag{B.59}$$

$$= -\Big[\sum_{i=1}^{V} x_{di} \frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} \boldsymbol{t}_k^\intercal \odot \exp\{\boldsymbol{\varsigma}_d\} \odot \tilde{\boldsymbol{\epsilon}}_{dr} \theta_{dkr} \Big] - \lambda \exp\{2\boldsymbol{\varsigma}_d\} + \mathbf{1} \tag{B.60}$$

$$\boxed{\nabla_{\boldsymbol{\varsigma}_d} \mathcal{L}_{\text{RP}}(q_d) = \mathbf{1} - \lambda \exp\{2\boldsymbol{\varsigma}_d\} - \Bigg[\Big(\sum_{i=1}^{V} x_{di} \Big) \Big(\frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{V} \theta_{dkr} \boldsymbol{t}_k^\intercal \odot \exp\{\boldsymbol{\varsigma}_d\} \odot \tilde{\boldsymbol{\epsilon}}_{dr} \Big) \Bigg].} \tag{B.61}$$

VB M-step: Updating model parameters:

Taking the derivative of complete objective Eq. (B.43) with respect to a row $\boldsymbol{t}_k$ from matrix $\boldsymbol{T}$.

$$\frac{\partial \mathcal{L}_{\mathsf{RP}}}{\partial \boldsymbol{t}_k} = \frac{\partial}{\partial \boldsymbol{t}_k} \sum_{d=1}^{D} \sum_{i=1}^{V} x_{di} \left[(m_i + \boldsymbol{t}_i \boldsymbol{\nu}_d) - \frac{1}{R} \sum_{r=1}^{R} \log \left(\sum_{j=1}^{V} \exp\{m_j + \boldsymbol{t}_j \, g(\, \boldsymbol{\epsilon}_{dr}\,)\} \right) \right] - \omega \sum_{i=1}^{V} ||\boldsymbol{t}_i||_1 \tag{B.62}$$

$$= \sum_{d=1}^{D} \left[x_{dk} \boldsymbol{\nu}_d^{\intercal} - \sum_{i=1}^{V} x_{di} \, \frac{1}{R} \sum_{r=1}^{R} g(\boldsymbol{\epsilon}_{dr})^{\intercal} \underbrace{\frac{\exp\{m_k + \boldsymbol{t}_k \, g(\boldsymbol{\epsilon}_{dr})\}}{\sum_j \exp\{m_j + \boldsymbol{t}_j \, g(\boldsymbol{\epsilon}_{dr})\}}}_{\theta_{dkr}} \right] - \omega \, \mathrm{sign}(\boldsymbol{t}_k) \tag{B.63}$$

$$= \sum_{d=1}^{D} \left[x_{dk} \boldsymbol{\nu}_d^{\intercal} - \sum_{i=1}^{V} x_{di} \, \frac{1}{R} \sum_{r=1}^{R} (\boldsymbol{\nu}_d + \exp\{\boldsymbol{\varsigma}_d\} \odot \tilde{\boldsymbol{\epsilon}}_{dr})^{\intercal} \, \theta_{dkr} \right] - \omega \, \mathrm{sign}(\boldsymbol{t}_k) \tag{B.64}$$

$$= \sum_{d=1}^{D} \left[x_{dk} \boldsymbol{\nu}_d^{\intercal} - (\sum_{i=1}^{V} x_{di}) \frac{1}{R} \sum_{r=1}^{R} \theta_{dkr} (\boldsymbol{\nu}_d^{\intercal} + \exp\{\boldsymbol{\varsigma}_d\} \odot \tilde{\boldsymbol{\epsilon}}_{dr}^{\intercal}) \right] - \omega \, \mathrm{sign}(\boldsymbol{t}_k) \tag{B.65}$$

$$\boxed{\nabla_{\boldsymbol{t}k} = \sum_{d=1}^{D} \left[x_{dk} \boldsymbol{\nu}_d^{\intercal} - \left[\Big(\sum_{i=1}^{V} x_{di} \Big) \, \frac{1}{R} \sum_{r=1}^{R} \theta_{dkr} \Big(\boldsymbol{\nu}_d^{\intercal} + \tilde{\boldsymbol{\epsilon}}_{dr}^{\intercal} \boldsymbol{L}_d^{\intercal} \Big) \right] \right] - \omega \, \mathrm{sign}(\boldsymbol{t}_k).} \tag{B.66}$$

Here, sign operates element-wise.

Appendix C

EM algorithm for Gaussian linear classifier with uncertainty

E-step: Obtaining the posterior distribution of latent variable $p(\boldsymbol{y}_d \,|\, \boldsymbol{\nu}_d, \Theta)$

.

Using the results from Petersen and Pedersen (2012) (Pg. 41, Eq. (358)):

$$\begin{aligned}
\log p(\boldsymbol{y}_d \mid \boldsymbol{\nu}_d) &= \log p(\boldsymbol{\nu}_d, \boldsymbol{y}_d) - \log p(\boldsymbol{\nu}_d) && \text{(C.1)}\\
&= \log p(\boldsymbol{\nu}_d \mid \boldsymbol{y}_d) + \log p(\boldsymbol{y}_d) - \log p(\boldsymbol{\nu}_d) && \text{(C.2)}\\
&= \log \mathcal{N}(\boldsymbol{\nu}_d \mid \boldsymbol{\mu}_d + \boldsymbol{y}_d, \boldsymbol{D}^{-1}) + \log \mathcal{N}(\boldsymbol{y}_d \mid \boldsymbol{0}, \boldsymbol{\Gamma}_d^{-1}) + \text{const} && \text{(C.3)}\\
&= -\frac{1}{2}(\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d))^{\intercal} \boldsymbol{D}(\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d)) - \frac{1}{2}\boldsymbol{y}_d^{\intercal}\boldsymbol{\Gamma}_d\boldsymbol{y}_d + \text{const} && \text{(C.4)}\\
&= -\frac{1}{2}(\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\intercal} \boldsymbol{D}(\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d)) - \frac{1}{2}\boldsymbol{y}_d^{\intercal}\boldsymbol{\Gamma}_d\boldsymbol{y}_d + \text{const} && \text{(C.5)}\\
&= \mathcal{N}(\boldsymbol{y}_d \mid \boldsymbol{u}_d, \boldsymbol{V}_d^{-1}) && \text{(C.6)}
\end{aligned}$$

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

$$\begin{aligned}
\boldsymbol{u}_d &= (\boldsymbol{D} + \boldsymbol{\Gamma}_d)^{-1}(\boldsymbol{D}(\boldsymbol{\nu}_d - \boldsymbol{\mu}_d) + \boldsymbol{\Gamma}_d\boldsymbol{0}) && \text{(C.7)}\\
&= [\boldsymbol{D}^{-1}(\boldsymbol{D} + \boldsymbol{\Gamma}_d)]^{-1}(\boldsymbol{\nu}_d - \boldsymbol{\mu}_d), && \text{(C.8)}
\end{aligned}$$

resulting in:

$$\boxed{\begin{aligned}
&\boldsymbol{u}_d = [\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{\Gamma}_d]^{-1}(\boldsymbol{\nu}_d - \boldsymbol{\mu}_d). && \text{(C.9)}\\
&\boldsymbol{V}_d = \boldsymbol{D} + \boldsymbol{\Gamma}_d. && \text{(C.10)}
\end{aligned}}$$

M-step: Maximizing the auxiliary function.

$$\begin{aligned}
\Theta^{\text{new}} &= \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{\text{old}}) && \text{(C.11)}\\
q(\boldsymbol{y}) &= p(\boldsymbol{y} \mid \boldsymbol{w}, \Theta^{\text{old}}) && \text{(C.12)}
\end{aligned}$$

Using the results from Petersen and Pedersen (2012) (Pg. 43, Eq. (378)), the auxiliary function $\mathcal{Q}(\Theta, \Theta^{\text{old}})$ is computed as:

$$\mathcal{Q}(\Theta, \Theta^{\text{old}}) = \mathbb{E}_q[\sum_{d}^{N} \log p(\boldsymbol{\nu}_d, \boldsymbol{y}_d)] \tag{C.13}$$

$$= \sum_{d}^{N} \mathbb{E}_q[\log p(\boldsymbol{\nu}_d \mid \boldsymbol{y}_d)] + \mathbb{E}_q[\log p(\boldsymbol{y}_d)] \tag{C.14}$$

$$= \sum_{d}^{N} \mathbb{E}_q[\log \mathcal{N}(\boldsymbol{\nu}_d \mid \boldsymbol{\mu}_d + \boldsymbol{y}_d, \boldsymbol{D}^{-1})] + \text{const} \tag{C.15}$$

$$= \frac{N}{2} \log \mid \boldsymbol{D} \mid - \frac{1}{2} \sum_{d}^{N} \Big[\mathbb{E}_q[(\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d))^{\intercal} \boldsymbol{D} (\boldsymbol{\nu}_d - (\boldsymbol{\mu}_d + \boldsymbol{y}_d))] \Big] + \text{const} \tag{C.16}$$

$$= \frac{N}{2} \log \mid \boldsymbol{D} \mid - \frac{1}{2} \sum_{d}^{N} \Big[\mathbb{E}_q[(\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\intercal} \boldsymbol{D} (\boldsymbol{y}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))] \Big] + \text{const} \tag{C.17}$$

$$= \frac{N}{2} \log \mid \boldsymbol{D} \mid - \frac{1}{2} \sum_{d}^{N} \Big[\text{Tr}(\boldsymbol{D} \boldsymbol{V}_d^{-1}) + (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\intercal} \boldsymbol{D} (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d)) \Big]. \tag{C.18}$$

Maximizing the auxiliary function $\mathcal{Q}$ w.r.t parameters $\Theta = \{\boldsymbol{M}, \boldsymbol{D}\}$:

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

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_\ell} = -\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\mu}_\ell} \sum_{d \in \mathcal{I}_\ell} \Big[(\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_\ell))^{\intercal} \boldsymbol{D} (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_l)) \Big] \tag{C.19}$$

$$= -\frac{1}{2} \sum_{d \in \mathcal{I}_\ell} 2 \boldsymbol{D} \left(\boldsymbol{\mu}_\ell - (\boldsymbol{\nu}_d - \boldsymbol{u}_d) \right) \tag{C.20}$$

$$= -\boldsymbol{D} \Big(\sum_{d \in \mathcal{I}_\ell} \boldsymbol{\mu}_\ell - \sum_{d \in \mathcal{I}_\ell} (\boldsymbol{\nu}_d - \boldsymbol{u}_d) \Big) \tag{C.21}$$

$$\boxed{\boldsymbol{\mu}_\ell = \frac{1}{|\mathcal{I}_\ell|} \sum_{n \in \mathcal{I}_\ell} (\boldsymbol{\nu}_d - \boldsymbol{u}_d).} \tag{C.22}$$

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

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{D}} = \frac{N}{2} \boldsymbol{D}^{-1} - \frac{1}{2} \Big(\sum_{d=1}^{D} \boldsymbol{V}_d^{-1} \Big)^{\intercal} - \frac{1}{2} \Big(\sum_{d=1}^{D} (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))(\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\intercal} \Big)^{\intercal} \tag{C.23}$$

$$\boxed{\boldsymbol{D}^{-1} = \frac{1}{D} \Big(\sum_{d=1}^{D} \boldsymbol{V}_d^{-1} + \sum_{d=1}^{D} (\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))(\boldsymbol{u}_d - (\boldsymbol{\nu}_d - \boldsymbol{\mu}_d))^{\intercal} \Big).} \tag{C.24}$$

Appendix D

Estimation of bias-corrected moments for adam optimization scheme

The equations for computing bias-corrected first and second order moment estimates of gradients is given here. For more details about ADAM optimization scheme, see Kingma and Ba (2015).

Let $\boldsymbol{g}_t$ denote a gradient of an objective function with-respect-to the desired parameter(s) at t^{th} step.

Let $\boldsymbol{f}_t$ and $\boldsymbol{s}_t$ denote first and second order moment estimates of gradient $\boldsymbol{g}_t$; where $\boldsymbol{f}_0$ and $\boldsymbol{s}_0$ are zero vectors.

The following formulae show the estimation of bias-corrected moments, as required by ADAM :

$$\boldsymbol{f}_{t+1} \leftarrow \beta_1 \boldsymbol{f}_t + (1-\beta_1)\boldsymbol{g}_t \tag{D.1}$$

$$\boldsymbol{s}_{t+1} \leftarrow \beta_2 \boldsymbol{s}_t + (1-\beta_2)\boldsymbol{g}_t^2 \tag{D.2}$$

$$\hat{\boldsymbol{f}}_{t+1} \leftarrow \frac{\boldsymbol{f}_{t+1}}{(1-\beta_1^{t+1})} \tag{D.3}$$

$$\hat{\boldsymbol{s}}_{t+1} \leftarrow \frac{\boldsymbol{s}_{t+1}}{(1-\beta_2^{t+1})} \tag{D.4}$$

where $\beta_1 = 0.9$, $\beta_2 = 0.999$, are the default exponential decay rates for the moment estimates, and, $\boldsymbol{g}_t^2$ denotes element-wise square, i.e., $\boldsymbol{g}_t \odot \boldsymbol{g}_t$.

Appendix E

Illustration of orthant-wise learning

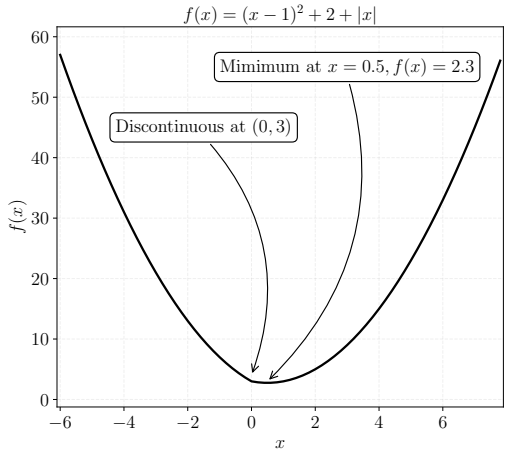
Illustration of Orthant-wise learning using an ℓ_1 regularized quadratic function:

$$f(x) = (x-1)^2 + 2 + |x| \tag{E.1}$$

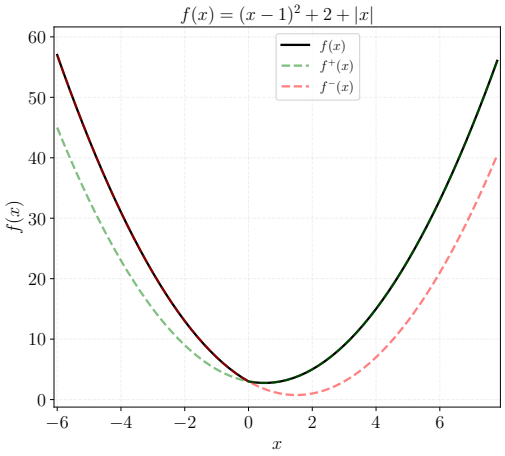
The positive and negative variants are

$$f^+(x) = (x-1)^2 + 2 + x \tag{E.2}$$
$$f^-(x) = (x-1)^2 + 2 - x, \tag{E.3}$$

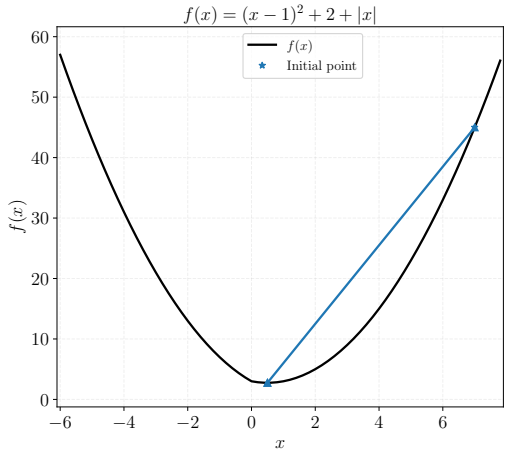
respectively. The sub-gradients are computed on these variants. They are in fact identical to the original $f(x)$ in one or the other quadrant. The illustrations are in Fig. E.1.



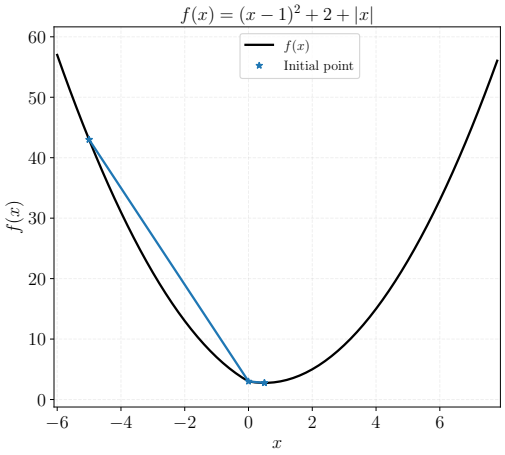
(a) ℓ_1 regularized quadratic function.



(b) The positive and negative variants of the objective function, that are used to compute subgradients.



(c) When the initial point is in the same quadrant as the minimum, single step using second order method leads to minimum.



(d) When the initial point is in different quadrant than the minima: Orthant-wise learning is employed and the updates do not cross the point of non-differentiability. Here two steps are needed to reach the minimum.

Figure E.1: Illustration of Orthant-wise learning using an ℓ_1 regularized quadratic function involving single variable.

The following example illustrate (Fig. E.2) Orthant-wise learning for an an ℓ_1 regularized quadratic function involving two variables

$$\mathcal{L} = (x-3)^2 + (y-4)^2 + 5(|x|+|y|). \tag{E.4}$$

The initial point is chosen to be in Quadrant 3, whereas the minimum is in Quadrant 1. The update steps following orthant-wise learning will stop at the points of non-differentiability. Once when crossing the line $y = 0$ and next when crossing the line $x = 0$.

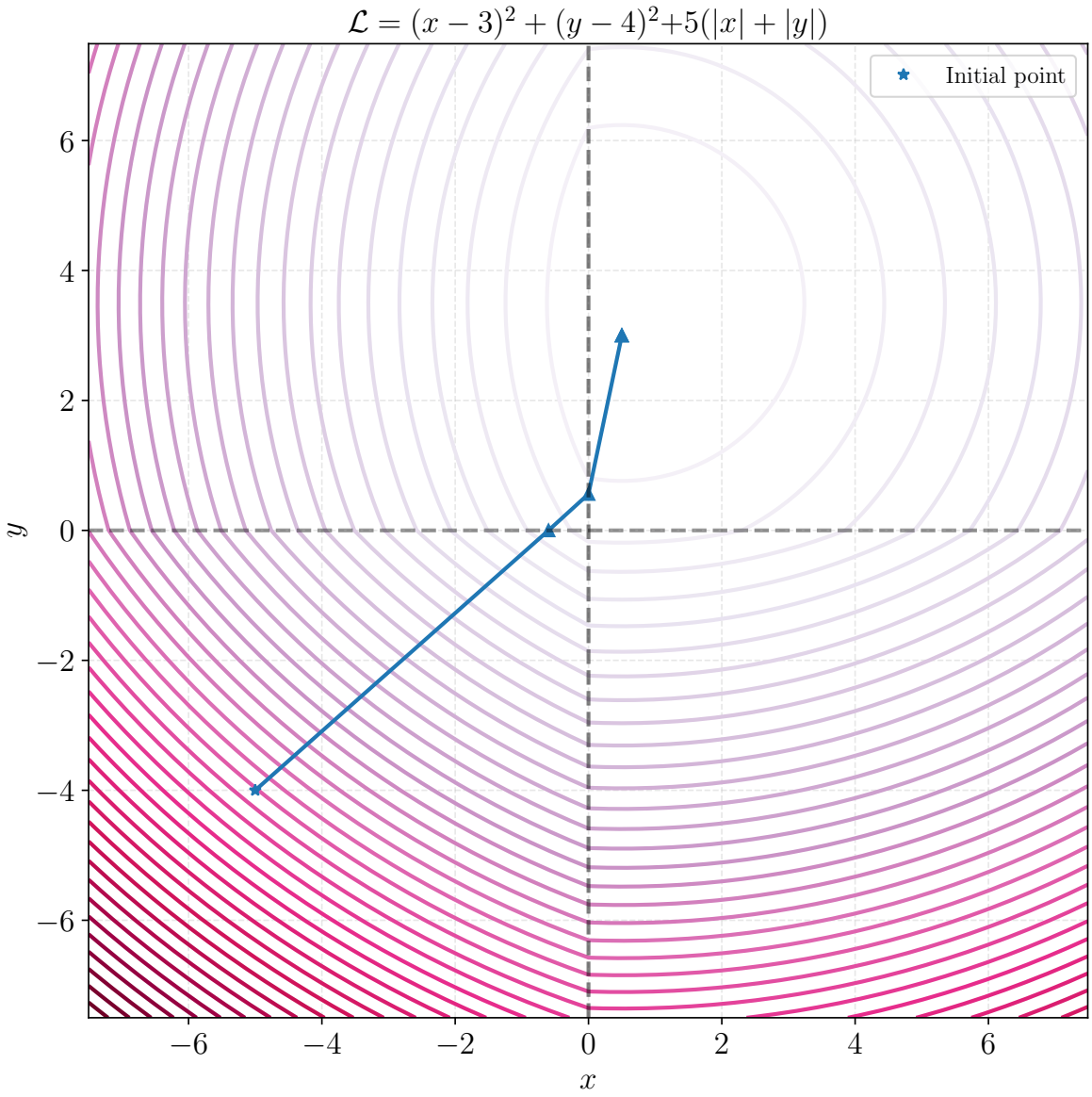


Figure E.2: Illustration orthant-wise learning for an ℓ_1 quadratic function involving two variables.

Appendix F

Datasets

F.1 20Newsgroups

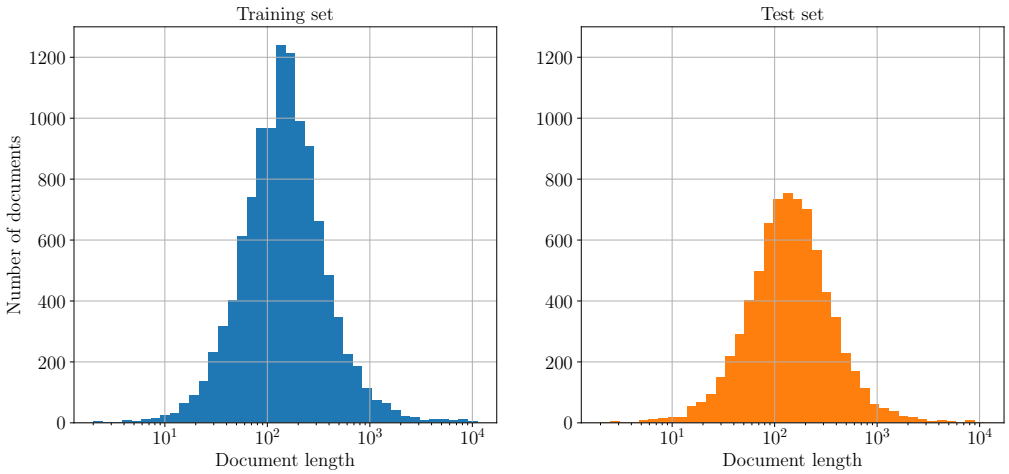
This is a freely available[1] standard text corpus, mainly used for topic identification and document clustering tasks. A standard preprocessed version (*20 Newsgroups*) is available that is usually preferred. It contains a total of 18774 documents with 61188 unique words comprising a closed set of 20 topics (see Table F.1). The training set consists of 11269 documents with a vocabulary of 53975 words and the test set consists of 7505 documents. The test set has 7213 words that are not present in training set and are ignored in our experiments.

Table F.1: Topics in *20 Newsgroups* dataset

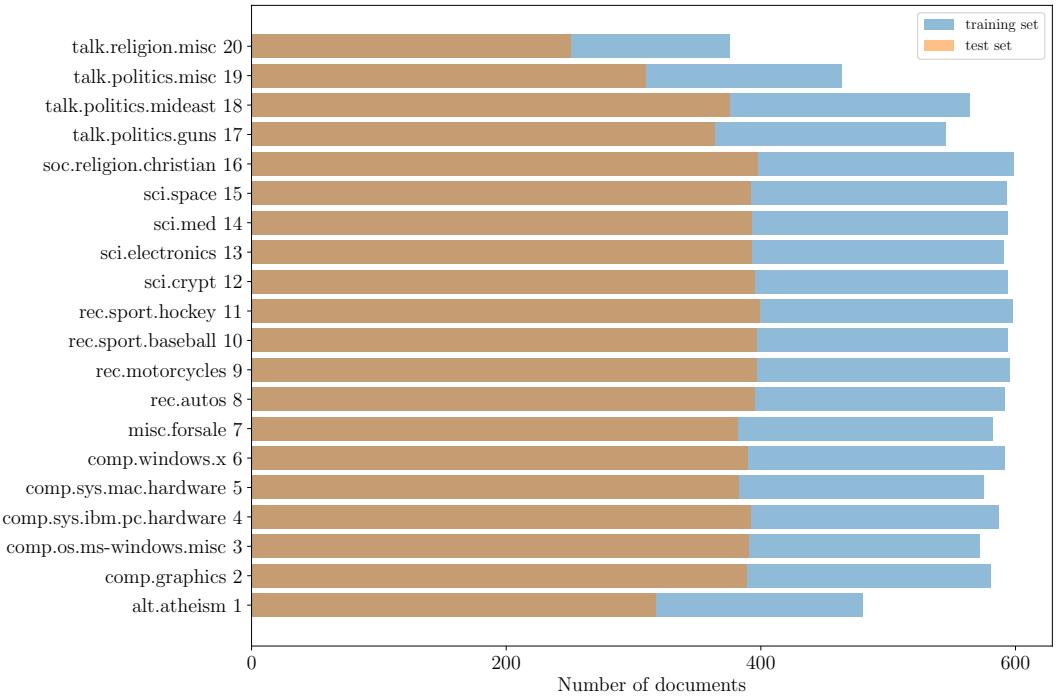
comp.graphics	rec.autos	sci.crypt
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med
comp.sys.mac.hardware	rec.sport.hockey	sci.space
comp.windows.x		
misc.forsale	talk.politics.misc	talk.religion.misc
	talk.politics.guns	alt.atheism
	talk.politics.mideast	soc.religion.christian

The Fig. F.1a shows the histogram of document lengths in training and test set and Fig. F.1b shows the training and test proportions per topic.

[1] http://qwone.com/~jason/20Newsgroups/



(a) Histogram of document lengths from training and test sets of *20Newsgroups* dataset.



(b) Number of training and test documents per topic

Figure F.1: 20 Newsgroups dataset

F.2 Fisher phase 1 speech corpus

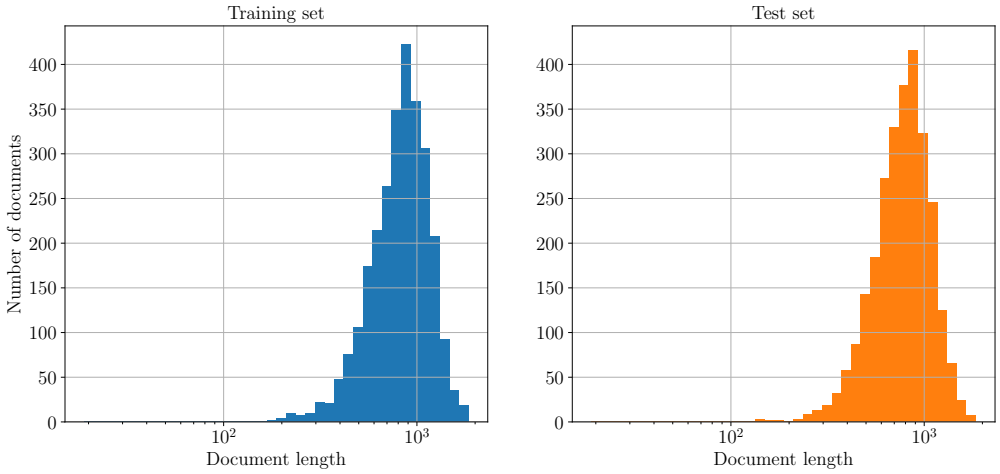
This is a collection of 5850 conversational telephone speech recordings with a closed set of 40 topics, and is distributed by Linguistic data consortium (LDC)[2]. Each conversation is approximately 10 minutes long with two sides of the call and is supposedly about one topic. While collecting the data, the callers were asked to talk about single topic, but sometimes they deviated. The details of data splits are presented in Table F.2; they are the same as used in earlier research Hazen et al. (2007); Hazen (2011); May et al. (2015). Our preprocessing involved removing punctuation and special characters, and we did not remove any stop words. The manual transcriptions are distributed by LDC[3], and the automatic ones are obtained from a DNN-HMM based automatic speech recognizer (ASR) system built using Kaldi toolkit Povey et al. (2011b) following the training algorithm (recipe) described in Veselý et al. (2013b). The ASR system resulted in 18% word-error-rate on a held-out test set. The vocabulary size while using manual transcriptions was 24854, for automatic, it was 18292, and the average document length (in number of words) is 830, and 856 respectively.

Table F.2: Data splits from *Fisher* phase 1 corpus, where each document represents one side of the conversation.

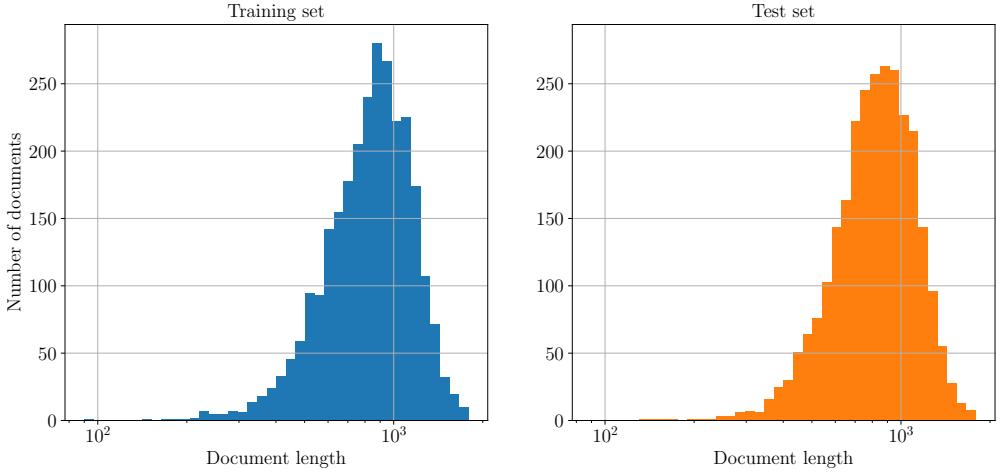
Set	**# docs.**	**Duration (hrs.)**
ASR training	6208	553
Topic ID training	2748	244
Topic ID test	2744	226

The histogram of document lengths (number of word tokens) is shown in Fig. F.2. The 40 topics along with their proportions in training and test sets is illustrated in Fig. F.3.

[2] https://catalog.ldc.upenn.edu/LDC2004S13
[3] https://catalog.ldc.upenn.edu/LDC2004T19



(a) Manual transcriptions.



(b) Automatic transcriptions.

Figure F.2: Histogram of document lengths from training and test sets of *Fisher* dataset.

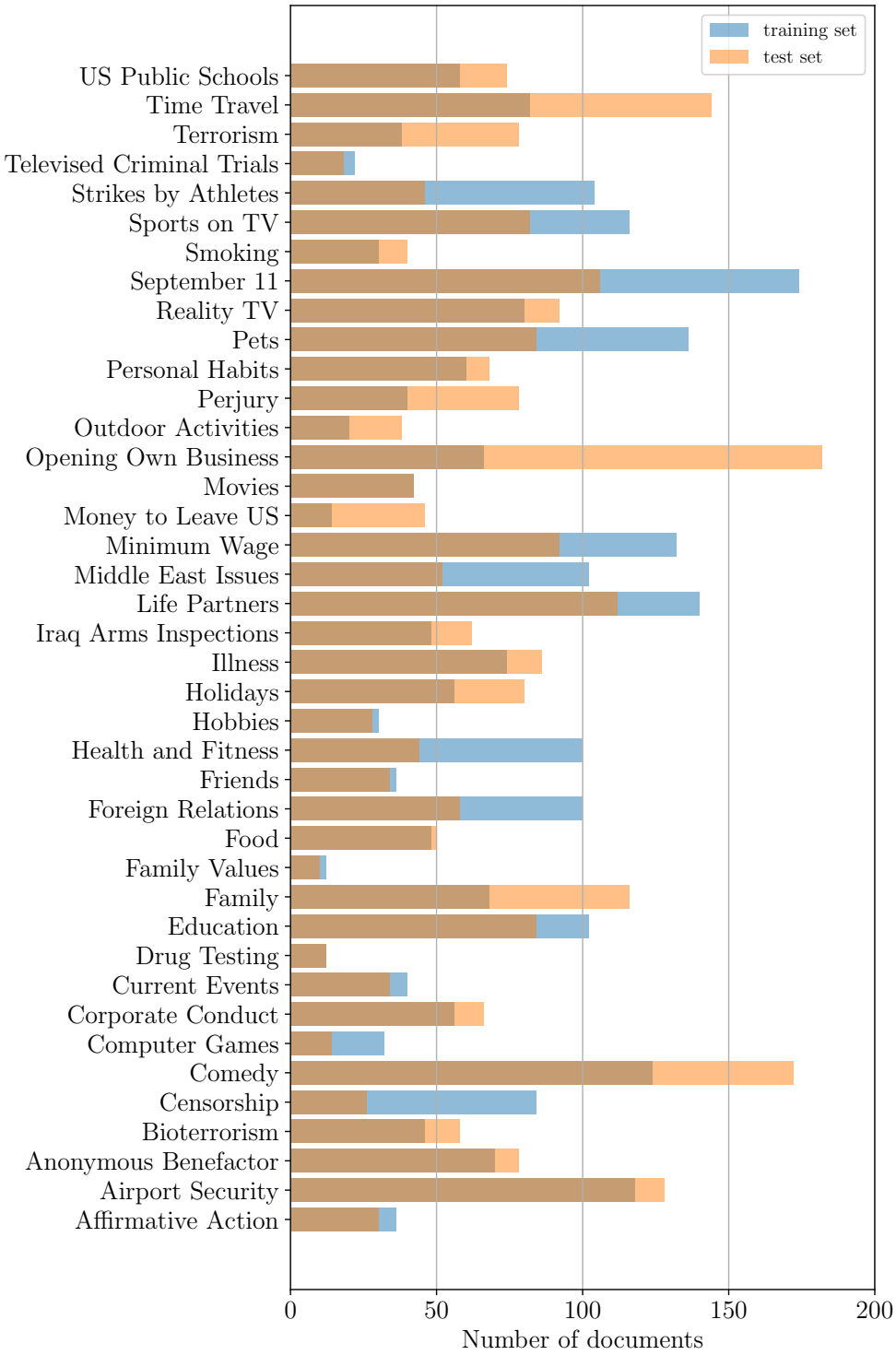


Figure F.3: Number of training and test documents per topic from *Fisher* dataset.

Bibliography

W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith. Massively Multilingual Word Embeddings. *CoRR*, abs/1602.01925, 2016. URL http://arxiv.org/abs/1602.01925.

G. Andrew and J. Gao. Scalable Training of L1-Regularized Log-Linear Models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, New York, USA, 2007. ACM. ISBN 978-1-59593-793-3. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2007/01/andrew07scalable.pdf.

M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *TACL*, 7:597–610, 2019. URL https://transacl.org/ojs/index.php/tacl/article/view/1742.

M. Artetxe, S. Ruder, and D. Yogatama. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL https://www.aclweb.org/anthology/2020.acl-main.421.

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 2018*, pages 3383–3387, 2018. doi: 10.21437/Interspeech.2018-1070. URL http://dx.doi.org/10.21437/Interspeech.2018-1070.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3(null):1137–1155, Mar. 2003. ISSN 1532-4435. URL http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, Apr. 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL http://doi.acm.org/10.1145/2133806.2133826.

D. M. Blei and J. D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems NIPS*, pages 147–154, December 2005. URL https://papers.nips.cc/paper/2906-correlated-topic-models.pdf.

D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1 (1):17–35, 2007. ISSN 19326157. URL http://www.jstor.org/stable/4537420.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. URL http://jmlr.org/papers/volume3/blei03a/blei03a.pdf.

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*, pages 349–356, 2018. doi: 10.21437/Odyssey.2018-49. URL http://dx.doi.org/10.21437/Odyssey.2018-49.

C. Chelba, X. Zhang, and K. Hall. Geo-location for voice search language modeling. In *Proc. Interspeech, ISCA*, pages 1438–1442, Sep 2015. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43817.pdf.

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*, pages 3511–3515, September 2015. URL https://www.isca-speech.org/archive/interspeech_2015/papers/i15_3511.pdf.

J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems 28, December 7-12 Montreal, Quebec, Canada*, pages 2980–2988, 2015. URL http://papers.nips.cc/paper/5653-a-recurrent-latent-variable-model-for-sequential-data.

S. Cumani, O. Plchot, and R. Fér. Exploiting i-vector posterior covariances for short-duration language recognition. In *Proceedings of Interspeech, ISCA*, pages 1002–1006, 2015. URL http://www.fit.vutbr.cz/research/view_pub.php.cs?id=10967.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the Americal Society for Information Science*, 41(6):391–407, 1990. URL http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf.

N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Trans. Audio, Speech & Language Processing*, 19(4):788–798, 2011. URL http://groups.csail.mit.edu/sls/archives/root/publications/2010/Dehak_IEEE_Transactions.pdf.

N. Depraetere and M. Vandebroek. A comparison of variational approximations for fast inference in mixed logit models. *Computational Statistics*, 32(1):93–125, 2017. URL https://link.springer.com/article/10.1007/s00180-015-0638-y.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Minneapolis, MN, USA, June 2-7, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. URL https://www.aclweb.org/anthology/N19-1423/.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011. URL http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse Additive Generative Models of Text. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, pages 1041–1048, USA, 2011. Omnipress. URL http://dl.acm.org/citation.cfm?id=3104482.3104613.

K. Ganchev, B. Taskar, F. Pereira, and J. ao Gama. Posterior vs parameter sparsity in latent variable models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 664–672. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3865-posterior-vs-parameter-sparsity-in-latent-variable-models.pdf.

M. Hannemann, J. Trmal, L. Ondel, S. Kesiraju, and L. Burget. Bayesian joint-sequence models for grapheme-to-phoneme conversion. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 2836–2840, March 2017.

T. J. Hazen. MCE Training Techniques for Topic Identification of Spoken Audio Documents. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2451–2460, Nov 2011. URL https://ieeexplore.ieee.org/document/5742980.

T. J. Hazen, F. Richardson, and A. Margolis. Topic Identification from Audio Recordings using Word and Phone Recognition Lattices. In *IEEE Workshop on ASRU*, pages 659–664, December 2007. URL https://ieeexplore.ieee.org/document/4430190.

M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, 2013. URL http://dl.acm.org/citation.cfm?id=2502622.

T. Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9. URL http://dl.acm.org/citation.cfm?id=2073796.2073829.

J. Howard and S. Ruder. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1031.

W. Jin, T. He, Y. Qian, and K. Yu. Paragraph Vector Based Topic Model for Language Model Adaptation. In *Proc. Interspeech*, pages 3516–3520, Sep 2015. URL https://isca-speech.org/archive/interspeech_2015/papers/i15_3516.pdf.

D. Jurafsky and J. H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA, 2009. ISBN 0131873210.

A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017. URL https://arxiv.org/abs/1703.04977.

P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel. PLDA for speaker verification with utterances of arbitrary duration. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7649–7653, May 2013. URL https://ieeexplore.ieee.org/document/6639151.

S. Kesiraju, G. Mantena, and K. Prahallad. IIIT-H System for MediaEval 2014 QUESST. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain*. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1263/mediaeval2014_submission_76.pdf.

S. Kesiraju, L. Burget, I. Szöke, and J. Černocký. Learning Document Representations Using Subspace Multinomial Model. In *Proceedings of Interspeech, ISCA*, pages 700–704, September 2016. doi: 10.21437/Interspeech.2016-1634. URL http://dx.doi.org/10.21437/Interspeech.2016-1634.

S. Kesiraju, R. Pappagari, L. Ondel, L. Burget, S. V. Gangashetty, et al. Topic identification of spoken documents using unsupervised acoustic unit discovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 5745–5749, March 2017. URL https://www.fit.vutbr.cz/research/groups/speech/publi/2017/kesiraju_icassp2017_0005745.pdf.

S. Kesiraju, O. Plchot, L. Burget, and S. V. Gangashetty. Learning document embeddings along with their uncertainties. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2319–2332, 2020a. doi: 10.1109/TASLP.2020.3012062.

S. Kesiraju, S. Sagar, O. Glembek, L. Burget, and S. V. Gangashetty. A Bayesian multilingual document model for zero-shot cross-lingual topic identification, 2020b. URL https://arxiv.org/abs/2007.01359.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6114.

M. Kockmann. *Subsapace Modeling of Prosodic Features for Speaker Verification*. PhD thesis, Brno University of Technology, 2011. URL https://www.fit.vut.cz/study/phd-thesis/228/.en.

M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Černocký. Prosodic speaker verification using subspace multinomial models with intersession compensation. In *Proceedings of Interspeech, ISCA*, pages 1061–1064, September 2010. URL http://www.fit.vutbr.cz/research/groups/speech/publi/2010/kockmann_interspeech2010_IS100048.pdf.

P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT. URL http://mt-archive.info/MTS-2005-Koehn.pdf.

S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 897–904, USA, 2008. ISBN 978-1-6056-0-949-2.

H. Larochelle and S. Lauly. A Neural Autoregressive Topic Model. In *Advances in NIPS*, pages 2717–2725, December 2012.

J. Lasserre. *Hybrid of Generative and Discriminative Methods for Machine Learning*. PhD thesis, University of Cambridge, March 2008.

Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the ICML*, pages 1188–1196, June 2014. URL https://dl.acm.org/citation.cfm?id=3045025.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, Dec. 2004. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1005332.1005345.

P. Liu, X. Qiu, and X. Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1284–1290. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832415.2832428.

A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning Word Vectors for Sentiment Analysis. In *The 49th Annual Meeting of the ACL: Human Language Technologies*, pages 142–150, June 2011. URL https://www.aclweb.org/anthology/P11-1015/.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. V. Durme. Topic identification and discovery on text and speech. In *Proceedings of the 2015 Conference on EMNLP*, pages 2377–2387, September 2015.

D. Mekala, V. Gupta, B. Paranjape, and H. Karnick. Scdv : Sparse composite document vectors using soft clustering over distributional representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 659–669, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1069. URL https://www.aclweb.org/anthology/D17-1069.

Y. Miao, L. Yu, and P. Blunsom. Neural Variational Inference for Text Processing. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 1727–1736. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045573.

T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *IEEE Spoken Language Technology Workshop*, pages 234–239, December 2012. URL https://ieeexplore.ieee.org/document/6424228.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, December 2013.

A. Mnih and K. Gregor. Neural Variational Inference and Learning in Belief Networks. In *Proceedings of the 31th ICML*, pages 1791–1799, June 2014.

O. Novotný, O. Plchot, O. Glembek, and L. Burget. Factorization of Discriminatively Trained i-Vector Extractor for Speaker Recognition. In *Proceedings of Interspeech*, pages 4330–4334. International Speech Communication Association, 2019. doi: 10.21437/Interspeech.2019-1757. URL https://www.fit.vut.cz/research/publication/12091.

L. Ondel, L. Burget, J. Černocký, and S. Kesiraju. Bayesian phonotactic language model for acoustic unit discovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 5750–5754, March 2017. URL http://www.fit.vutbr.cz/research/groups/speech/publi/2017/hannemann_icassp2017_0002836.pdf.

L. Ondel, K. H. Vydana, L. Burget, and J. Černocký. Bayesian Subspace Hidden Markov Model for Acoustic Unit Discovery. In *Proceedings of Interspeech*, pages 261–265. International Speech Communication Association, 2019. doi: 10.21437/Interspeech.2019-2224. URL https://www.fit.vut.cz/research/publication/12084.

J. Paisley, D. M. Blei, and M. I. Jordan. Variational Bayesian Inference with Stochastic Search. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12, page 1363–1370, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

R. Pappagari, J. Villalba, and N. Dehak. Joint verification-identification in end-to-end multi-scale cnn framework for topic identification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203, April 2018. doi: 10.1109/ICASSP.2018.8461673.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Workshop*, 2017.

J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, ACL*, pages 1532–1543, October 2014. URL https://www.aclweb.org/anthology/D14-1162.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

K. B. Petersen and M. S. Pedersen. The Matrix Cookbook, Nov 2012.

O. Plchot, P. Matějka, R. Fér, O. Glembek, O. Novotný, J. Pešán, K. Veselý, L. Ondel, M. Karafiát, F. Grézl, S. Kesiraju, L. Burget, N. Brummer, P. du Albert Swart, S. Cumani, H. S. Mallidi, and R. Li. BAT System Description for NIST LRE 2015. In *Odyssey - The Speaker and Language Recognition Workshop, ISCA*, pages 166–173, June 2016.

O. Plchot, P. Matejka, O. Novotnỳ, S. Cumani, A. Lozano-Diez, J. Slavicek, M. Diez, F. Grézl, O. Glembek, K. Mounika, A. Silnova, L. Burget, L. Ondel, S. Kesiraju, and J. Rohdin. Analysis of BUT-PT Submission for NIST LRE 2017. In *Odyssey - The Speaker and Language Recognition Workshop, ISCA*, pages 47–53, June 2018.

D. Povey. *SUBSPACE GAUSSIAN MIXTURE MODELS FOR SPEECH RECOGNITION*, 2009. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2009/05/ubmdoc.pdf.

D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas. The subspace gaussian mixture model - A structured model for speech recognition. *Computer Speech & Language*, 25(2):404–439, 2011a.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi Speech Recognition Toolkit. In *IEEE Workshop on ASRU*. IEEE Signal Processing Society, Dec 2011b.

K. Prahallad, A. Vadapalli, S. Kesiraju, H. A. Murthy, S. Lata, T. Nagarajan, M. Prasanna, H. Patil, A. K. Sao, S. King, A. W. Black, and K. Tokuda. The Blizzard Challenge 2014. In *Proceedings of Blizzard Workshop*, 2014.

B. Pulugundla, M. K. Baskar, S. Kesiraju, E. Egorova, M. Karafiát, L. Burget, and J. Černocký. BUT System for Low Resource Indian Language ASR. In *Proc. Interspeech 2018*, pages 3182–3186, 2018. doi: 10.21437/Interspeech.2018-1302. URL http://dx.doi.org/10.21437/Interspeech.2018-1302.

A. Radford. Improving language understanding by generative pre-training. 2018.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

S. Ruder, I. Vulić, and A. Søgaard. A Survey of Cross-lingual Word Embedding Models. *J. Artif. Int. Res.*, 65(1):569–630, May 2019. ISSN 1076-9757. doi: 10.1613/jair.1.11640. URL https://doi.org/10.1613/jair.1.11640.

M. Schmidt. *Graphical Model Structure Learning with ℓ_1 Regularization*. PhD thesis, The University of British Columbia, August 2010.

H. Schwenk and M. Douze. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 157–167, 2017. URL https://www.aclweb.org/anthology/W17-2619/.

H. Schwenk and X. Li. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC*

2018, Miyazaki, Japan, May 7-12, 2018., 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/658.html.

M. V. S. Shashanka, B. Raj, and P. Smaragdis. Sparse Overcomplete Latent Variable Decomposition of Counts Data. In *NIPS*, pages 1313–1320, December 2007.

A. Siddhant, M. Johnson, H. Tsai, N. Arivazhagan, J. Riesa, A. Bapna, O. Firat, and K. Raman. Evaluating the Cross-Lingual Effectiveness of Massively Multilingual Neural Machine Translation, 2019.

M. Soufifar. *Subspace Modeling of Discrete Features for Language Recognition*. PhD thesis, Norwegian University of Science and Technology, November 2014.

M. Soufifar, M. Kockmann, L. Burget, et al. iVector Approach to Phonotactic Language Recognition. In *Proceedings of Interspeech, ISCA*, pages 2913–2916, August 2011.

M. Soufifar, L. Burget, O. Plchot, et al. Regularized Subspace n-Gram Model for Phonotactic iVector Extraction. In *Proc. Interspeech, ISCA*, pages 74–78, August 2013.

A. Srivastava and C. A. Sutton. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=BybtVK9lg.

N. Srivastava, R. Salakhutdinov, and G. E.Hinton. Modeling Documents with Deep Boltzmann Machines. In *UAI*, August 2013a.

N. Srivastava, R. Salakhutdinov, and G. Hinton. Modeling documents with a deep boltzmann machine. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 616–624, Arlington, Virginia, United States, 2013b. AUAI Press. URL http://dl.acm.org/citation.cfm?id=3023638.3023701.

C. Sun, H. Yan, X. Qiu, and X. Huang. Gaussian Word Embedding with a Wasserstein Distance Loss. *ArXiv*, 1808.07016v7, 2018. URL https://arxiv.org/pdf/1808.07016.pdf.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

K. Veselý, A. Ghoshal, L. Burget, and D. Povey. Sequence-discriminative training of deep neural networks. In *Proceedings of Interspeech, ISCA*, pages 2345–2349, August 2013a. URL https://www.isca-speech.org/archive/interspeech_2013/i13_2345.html.

K. Veselý, A. Ghoshal, L. Burget, and D. Povey. Sequence-discriminative training of deep neural networks. In *Proc. Interspeech*, pages 2345–2349, August 2013b.

L. Vilnis and A. McCallum. Word representations via gaussian embedding. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6623.

H. M. Wallach. Topic Modeling: Beyond Bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143967. URL http://doi.acm.org/10.1145/1143844.1143967.

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*, pages 178–185, August 2006.

J. Wintrode and S. Khudanpur. Limited resource term detection for effective topic identification of speech. In *IEEE ICASSP*, pages 7118–7122, May 2014.

S. Wu and M. Dredze. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China, nov 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://www.aclweb.org/anthology/D19-1077.

Y. Xiao and W. Y. Wang. Quantifying Uncertainties in Natural Language Processing Tasks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI Honolulu, Hawaii, USA, January 27 - February 1*, pages 7322–7329, 2019. doi: 10.1609/aaai.v33i01.33017322. URL https://doi.org/10.1609/aaai.v33i01.33017322.

W. Xu, X. Liu, and Y. Gong. Document Clustering Based on Non-negative Matrix Factorization. In *SIGIR*, pages 267–273, New York, USA, 2003. ACM.

Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical attention networks for document classification. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17*, pages 1480–1489, 2016a. URL https://www.aclweb.org/anthology/N16-1174/.

Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical Attention Networks for Document Classification. In *NAACL HLT, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, June 2016b. URL http://aclweb.org/anthology/N/N16/N16-1174.pdf.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969312.

J. Zhu and E. P. Xing. Sparse Topical Coding. In *Proceedings of the 27th Conference on UAI*, pages 831–838, July 2011. URL https://dl.acm.org/citation.cfm?id=3020644.