Gaussian LDA for Topic Models with Word Embeddings

Rajarshi Das*, **Manzil Zaheer***, **Chris Dyer**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{rajarshd, manzilz, cdyer} @cs.cmu.edu

Abstract

Continuous space word embeddings learned from large, unstructured corpora have been shown to be effective at capturing semantic regularities in language. In this paper we replace LDA's parameterization of "topics" as categorical distributions over opaque word types with multivariate Gaussian distributions on the embedding space. This encourages the model to group words that are *a priori* known to be semantically related into topics. To perform inference, we introduce a fast collapsed Gibbs sampling algorithm based on Cholesky decompositions of covariance matrices of the posterior predictive distributions. We further derive a scalable algorithm that draws samples from stale posterior predictive distributions and corrects them with a Metropolis–Hastings step. Using vectors learned from a domain-general corpus (English Wikipedia), we report results on two document collections (20-newsgroups and NIPS). Qualitatively, Gaussian LDA infers different (but still very sensible) topics relative to standard LDA. Quantitatively, our technique outperforms existing models at dealing with OOV words in held-out documents.

1 Introduction

Latent Dirichlet Allocation (LDA) is a Bayesian technique that is widely used for inferring the topic structure in corpora of documents. It conceives of a document as a mixture of a small number of topics, and topics as a (relatively sparse) distribution over word types (Blei et al., 2003). These priors are remarkably effective at producing useful results. However, our intuitions tell us that while documents may indeed be conceived of as a mixture of topics, we should further expect topics to be *semantically coherent*. Indeed, standard human evaluations of topic modeling performance are designed to elicit assessment of semantic coherence (Chang et al., 2009; Newman et al., 2009). However, this prior preference for semantic coherence is not encoded in the model, and any such observation of semantic coherence found in the inferred topic distributions is, in some sense, accidental. In this paper, we develop a variant of LDA that operates on continuous space embeddings of words—rather than word types—to impose a prior expectation for semantic coherence. Our approach replaces the opaque word types usually modeled in LDA with continuous space embeddings of these words, which are generated as draws from a multivariate Gaussian.

How does this capture our preference for semantic coherence? Word embeddings have been shown to capture lexico-semantic regularities in language: words with similar syntactic and semantic properties are found to be close to each other in the embedding space (Agirre et al., 2009; Mikolov et al., 2013). Since Gaussian distributions capture a notion of centrality in space, and semantically related words are localized in space, our Gaussian LDA model encodes a prior preference for semantically coherent topics. Our model further has several advantages. Traditional LDA assumes a fixed vocabulary of word types. This modeling assumption drawback as it cannot handle *out of vocabulary* (OOV) words in "held out" documents. Zhai and Boyd-Graber (2013) proposed an approach to address this problem by drawing topics from a Dirichlet Process with a base distribution over all possible character strings (i.e., words). While this model can in principle handle unseen words, the only bias toward being included in a particular topic comes from the topic assignments in the rest

*Both student authors had equal contribution.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 795–804, Beijing, China, July 26-31, 2015.

of the document. Our model can exploit the contiguity of semantically similar words in the embedding space and can assign high topic probability to a word which is similar to an existing topical word even if it has never been seen before.

The main contributions of our paper are as follows: We propose a new technique for topic modeling by treating the document as a collection of word embeddings and topics itself as multivariate Gaussian distributions in the embedding space (§3). We explore several strategies for collapsed Gibbs sampling and derive scalable algorithms, achieving asymptotic speed-up over the naïve implementation (§4). We qualitatively show that our topics make intuitive sense and quantitatively demonstrate that our model captures a better representation of a document in the topic space by outperforming other models in a classification task (§5).

2 Background

Before going to the details of our model we provide some background on two topics relevant to our work: vector space word embeddings and LDA.

2.1 Vector Space Semantics

According to the **distributional hypothesis** (Harris, 1954), words occurring in similar contexts tend to have similar meaning. This has given rise to data-driven learning of word vectors that capture lexical and semantic properties, which is now a technique of central importance in natural language processing. These word vectors can be used for identifying semantically related word pairs (Turney, 2006; Agirre et al., 2009) or as features in downstream text processing applications (Turian et al., 2010; Guo et al., 2014). Word vectors can either be constructed using low rank approximations of cooccurrence statistics (Deerwester et al., 1990) or using internal representations from neural network models of word sequences (Collobert and Weston, 2008). We use a recently popular and fast tool called `word2vec`[1], to generate skip-gram word embeddings from unlabeled corpus. In this model, a word is used as an input to a log-linear classifier with continuous projection layer and words within a certain window before and after the words are predicted.

[1] `https://code.google.com/p/word2vec/`

2.2 Latent Dirichlet Allocation (LDA)

LDA (Blei et al., 2003) is a probabilistic topic model of corpora of documents which seeks to represent the underlying thematic structure of the document collection. They have emerged as a powerful new technique of finding useful structure in an unstructured collection as it learns distributions over words. The high probability words in each distribution gives us a way of understanding the contents of the corpus at a very high level. In LDA, each document of the corpus is assumed to have a distribution over K topics, where the discrete topic distributions are drawn from a symmetric dirichlet distribution. The generative process is as follows.

1. for k = 1 to K
   (a) Choose topic $\boldsymbol{\beta}_k \sim \mathsf{Dir}(\boldsymbol{\eta})$
2. for each document d in corpus D
   (a) Choose a topic distribution $\boldsymbol{\theta}_d \sim \mathsf{Dir}(\boldsymbol{\alpha})$
   (b) for each word index n from 1 to N_d
      i. Choose a topic $z_n \sim \mathsf{Categorical}(\boldsymbol{\theta}_d)$
      ii. Choose word $w_n \sim \mathsf{Categorical}(\boldsymbol{\beta}_{z_n})$

As it follows from the definition above, a topic is a discrete distribution over a fixed vocabulary of word types. This modeling assumption precludes new words to be added to topics. However modeling topics as a continuous distribution over word embeddings gives us a way to address this problem. In the next section we describe Gaussian LDA, a straightforward extension of LDA that replaces categorical distributions over word types with multivariate Gaussian distributions over the word embedding space.

3 Gaussian LDA

As with multinomial LDA, we are interested in modeling a collection of documents. However, we assume that rather than consisting of sequences of word types, documents consist of sequences of word embeddings. We write $\mathbf{v}(w) \in \mathbb{R}^M$ as the embedding of word of type w or $\mathbf{v}_{d,i}$ when we are indexing a vector in a document d at position i.

Since our observations are no longer discrete values but continuous vectors in an M-dimensional space, we characterize each topic k as a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. The choice of a Gaussian parameterization is justified by both analytic convenience and observations that Euclidean distances

$$p(z_{d,i} = k \mid \boldsymbol{z}_{-(d,i)}, \mathbf{V}_d, \boldsymbol{\zeta}, \boldsymbol{\alpha}) \propto (n_{k,d} + \alpha_k) \times t_{\nu_k - M + 1}\left(\mathbf{v}_{d,i} \,\middle|\, \boldsymbol{\mu}_k, \frac{\kappa_k + 1}{\kappa_k}\boldsymbol{\Sigma}_k\right) \quad (1)$$

Figure 1: Sampling equation for the collapsed Gibbs sampler; refer to text for a description of the notation.

between embeddings correlate with semantic similarity (Collobert and Weston, 2008; Turney and Pantel, 2010; Hermann and Blunsom, 2014). We place conjugate priors on these values: a Gaussian centered at zero for the mean and an inverse Wishart distribution for the covariance. As before, each document is seen as a mixture of topics whose proportions are drawn from a symmetric Dirichlet prior. The generative process can thus be summarized as follows:

1. for k = 1 to K
   (a) Draw topic covariance $\boldsymbol{\Sigma}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}, \nu)$
   (b) Draw topic mean $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\kappa}\boldsymbol{\Sigma}_k)$
2. for each document d in corpus D
   (a) Draw topic distribution $\boldsymbol{\theta}_d \sim \mathsf{Dir}(\boldsymbol{\alpha})$
   (b) for each word index n from 1 to N_d
      i. Draw a topic $z_n \sim \mathsf{Categorical}(\boldsymbol{\theta}_d)$
      ii. Draw $\mathbf{v}_{d,n} \sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$

This model has previously been proposed for obtaining indexing representations for audio retrieval (Hu et al., 2012). They use variational/EM method for posterior inference. Although we don't do any experiment to compare the running time of both approaches, the per-iteration computational complexity is same for both inference methods. We propose a faster inference technique using Cholesky decomposition of covariance matrices which can be applied to both the Gibbs and variational/EM method. However we are not aware of any straightforward way of applying the aliasing trick proposed by (Li et al., 2014) on the variational/EM method which gave us huge improvement on running time (see Figure 2). Another work which combines embedding with topic models is by (Wan et al., 2012) where they jointly learn the parameters of a neural network and a topic model to capture the topic distribution of low dimensional representation of images.

4 Posterior Inference

In our application, we observe documents consisting of word vectors and wish to infer the posterior distribution over the topic parameters, proportions, and the topic assignments of individual words. Since there is no analytic form of the posterior, approximations are required. Because of our choice of conjugate priors for topic parameters and proportions, these variables can be analytically integrated out, and we can derive a collapsed Gibbs sampler that resamples topic assignments to individual word vectors, similar to the collapsed sampling scheme proposed by Griffiths and Steyvers (2004).

The conditional distribution we need for sampling is shown in Figure 1. Here, $\boldsymbol{z}_{-(d,i)}$ represents the topic assignments of all word embeddings, excluding the one at i^{th} position of document d; $\mathbf{V}_d$ is the sequence of vectors for document d; $t_{\nu'}(\mathbf{x} \mid \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ is the multivariate t - distribution with ν' degrees of freedom and parameters $\boldsymbol{\mu}'$ and $\boldsymbol{\Sigma}'$. The tuple $\boldsymbol{\zeta} = (\boldsymbol{\mu}, \kappa, \boldsymbol{\Sigma}, \nu)$ represents the parameters of the prior distribution.

It should be noted that the first part of the equation which expresses the probability of topic k in document d is the same as that of LDA. This is because the portion of the model which generates a topic for each word (vector) from its document topic distribution is still the same. The second part of the equation which expresses the probability of assignment of topic k to the word vector $\mathbf{v}_{d,i}$ given the current topic assignments (aka posterior predictive) is given by a multivariate t distribution with parameters $(\boldsymbol{\mu}_k, \kappa_k, \boldsymbol{\Sigma}_k, \nu_k)$. The parameters of the posterior predictive distribution are given as (Murphy, 2012):

$$\begin{aligned} \kappa_k &= \kappa + N_k \qquad \boldsymbol{\mu}_k = \frac{\kappa\boldsymbol{\mu} + N_k\bar{\mathbf{v}}_k}{\kappa_k} \\ \nu_k &= \nu + N_k \qquad \boldsymbol{\Sigma}_k = \frac{\boldsymbol{\Psi}_k}{(\nu_k - M + 1)} \\ \boldsymbol{\Psi}_k &= \boldsymbol{\Psi} + \mathbf{C}_k + \frac{\kappa N_k}{\kappa_k}(\bar{\mathbf{v}}_k - \boldsymbol{\mu})(\bar{\mathbf{v}}_k - \boldsymbol{\mu})^\top \end{aligned} \quad (2)$$

where $\bar{\mathbf{v}}_k$ and $\mathbf{C}_k$ are given by,

$$\bar{\mathbf{v}}_k = \frac{\sum_d \sum_{i:z_{d,i}=k} (\mathbf{v}_{d,i})}{N_k}$$
$$\mathbf{C}_k = \sum_d \sum_{i:z_{d,i}=k} (\mathbf{v}_{d,i} - \bar{\mathbf{v}}_k)(\mathbf{v}_{d,i} - \bar{\mathbf{v}}_k)^\top$$

Here $\bar{\mathbf{v}}_k$ is the sample mean and $\mathbf{C}_k$ is the scaled form of sample covariance of the vectors with topic assignment k. N_k represents the count of words assigned to topic k across all documents. Intuitively the parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ represents the posterior mean and covariance of the topic distribution and κ_k, ν_k represents the strength of the prior for mean and covariance respectively.

Analysis of running time complexity

As can be seen from (1), for computation of the posterior predictive we need to evaluate the determinant and inverse of the posterior covariance matrix. Direct naïve computation of these terms require $O(M^3)$ operations. Moreover, during sampling as words get assigned to different topics, the parameters $(\boldsymbol{\mu}_k, \kappa_k, \boldsymbol{\Psi}_k, \nu_k)$ associated with a topic changes and hence we have to recompute the determinant and inverse matrix. Since these step has to be recomputed several times (as many times as number of words times number of topics in one Gibbs sweep, in the worst case), it is critical to make the process as efficient as possible. We speed up this process by employing a combination of modern computational techniques and mathematical (linear algebra) tricks, as described in the following subsections.

4.1 Faster sampling using Cholesky decomposition of covariance matrix

Having another look at the posterior equation for $\boldsymbol{\Psi}_k$, we can re-write the equation as:

$$\begin{aligned}\boldsymbol{\Psi}_k &= \boldsymbol{\Psi} + \mathbf{C}_k + \frac{\kappa N_k}{\kappa_k}(\bar{\mathbf{v}}_k - \boldsymbol{\mu})(\bar{\mathbf{v}}_k - \boldsymbol{\mu})^\top \\ &= \boldsymbol{\Psi} + \sum_d \sum_{i:z_{d,i}=k} \mathbf{v}_{d,i}\mathbf{v}_{d,i}^\top - \kappa_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\ &\quad + \kappa \boldsymbol{\mu}\boldsymbol{\mu}^\top. \end{aligned} \tag{3}$$

During sampling when we are computing the assignment probability of topic k to $\mathbf{v}_{d,i}$, we need to calculate the updated parameters of the topic. Using (3) it can be shown that $\boldsymbol{\Psi}_k$ can be updated from current value of $\boldsymbol{\Psi}_k$, after updating $\kappa_k.\nu_k$ and μ_k, as follows:

$$\boldsymbol{\Psi}_k \leftarrow \boldsymbol{\Psi}_k + \frac{\kappa_k}{\kappa_k - 1}(\boldsymbol{\mu}_k - \mathbf{v}_{d,i})(\boldsymbol{\mu}_k - \mathbf{v}_{d,i})^\top. \tag{4}$$

This equation has the form of a rank 1 update, hinting towards use of Cholesky decomposition. If we have the Cholesky decomposition of $\boldsymbol{\Psi}_k$ computed, then we have tools to update $\boldsymbol{\Psi}_k$ cheaply. Since $\boldsymbol{\Psi}_k$ and $\boldsymbol{\Sigma}_k$ are off by only a scalar factor, we can equivalently talk about $\boldsymbol{\Sigma}_k$. Equation (4) can also be understood in the following way. During sampling, when a word embedding $\mathbf{v}_{d,i}$ gets a new assignment to a topic, say k, then the new value of the topic covariance can be computed from the current one using just a rank 1 update.[2] We next describe how to exploit the Cholesky decomposition representation to speed up computations.

For sake of completeness, any symmetric $M \times M$ real matrix $\boldsymbol{\Sigma}_k$ is said to be positive definite if $\forall \mathbf{z} \in \mathbb{R}^M : \mathbf{z}^\top \boldsymbol{\Sigma}_k \mathbf{z} > 0$. The Cholesky decomposition of such a symmetric positive definite matrix $\boldsymbol{\Sigma}_k$ is nothing but its decomposition into the product of some lower triangular matrix $\mathbf{L}$ and its transpose, i.e.

$$\boldsymbol{\Sigma}_k = \mathbf{L}\mathbf{L}^\top.$$

Finding this factorization also take cubic operation. However given Cholesky decomposition of $\boldsymbol{\Sigma}_k$, after a rank 1 update (or downdate), i.e. the operation:

$$\boldsymbol{\Sigma}_k \leftarrow \boldsymbol{\Sigma}_k + \mathbf{z}\mathbf{z}^\top$$

we can find the factorization of new $\boldsymbol{\Sigma}_k$ in just quadratic time (Stewart, 1998). We will use this trick to speed up the computations[3]. Basically, instead of computing determinant and inverse again in cubic time, we will use such rank 1 update (downdate) to find new determinant and inverse in an efficient manner as explained in details below.

To compute the density of the posterior predictive $t-$distibution, we need to compute the determinant $|\boldsymbol{\Sigma}_k|$ and the term of the form $(\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)$. The Cholesky decomposition of the covariance matrix can be used for efficient computation of these expression as shown below.

[2] Similarly the covariance of the old topic assignment of the word w can be computed using a rank 1 downdate

[3] For our experiments, we set the prior covariance to be 3*$\mathcal{I}$, which is a positive definite matrix.

Computation of determinant: The determinant of $\mathbf{\Sigma}_k$ can be computed from from its Cholesky decomposition $\mathbf{L}$ as:

$$\log(|\mathbf{\Sigma}_k|) = 2 \times \sum_{i=1}^{M} \log\left(\mathbf{L}_{i,i}\right).$$

This takes linear time in the order of dimension and is clearly a significant gain from cubic time complexity.

Computation of $(\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)^\top \mathbf{\Sigma}_k^{-1} (\mathbf{v}_{d,i} - \boldsymbol{\mu})$: Let $\mathbf{b} = (\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)$. Now $\mathbf{b}^\top \mathbf{\Sigma}^{-1} \mathbf{b}$ can be written as

$$\begin{aligned}\mathbf{b}^\top \mathbf{\Sigma}^{-1} \mathbf{b} &= \mathbf{b}^\top (\mathbf{L}\mathbf{L}^\top)^{-1} \mathbf{b} \\ &= \mathbf{b}^T (\mathbf{L}^{-1})^\top \mathbf{L}^{-1} \mathbf{b} \\ &= (\mathbf{L}^{-1}\mathbf{b})^\top (\mathbf{L}^{-1}\mathbf{b})\end{aligned}$$

Now $(\mathbf{L}^{-1}\mathbf{b})$ is the solution of the equation $\mathbf{L}\mathbf{x} = \mathbf{b}$. Also since $\mathbf{L}$ is a lower triangular matrix, this equation can be solved easily using forward substitution. Lastly we will have to take an inner product of $\mathbf{x}$ and $\mathbf{x}^\top$ to get the value of $(\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)^\top \mathbf{\Sigma}^{-1} (\mathbf{v}_{d,i} - \boldsymbol{\mu}_k)$. This step again takes quadratic time and is again a savings from the cubic time complexity.

4.2 Further reduction of sampling complexity using Alias Sampling

Although Cholesky trick helps us to reduce the sampling complexity of a embedding to $O(KM^2)$, it can still be impractical.In Gaussian LDA, the Gibbs sampling equation (1) can be split into two terms. The first term $n_{k,d} \times t_{\nu_k - M + 1}\left(\mathbf{v}_{d,i} \middle| \boldsymbol{\mu}_k, \frac{\kappa_k + 1}{\kappa_k} \mathbf{\Sigma}_k\right)$ denotes the document contribution and the second term $\alpha_k \times t_{\nu_k - M + 1}\left(\mathbf{v}_{d,i} \middle| \boldsymbol{\mu}_k, \frac{\kappa_k + 1}{\kappa_k} \mathbf{\Sigma}_k\right)$ denotes the language model contribution. Empirically one can make two observations about these terms. First, $n_{k,d}$ is often a sparse vector, as a document most likely contains only a few of the topics. Secondly, topic parameters (μ_k, Σ_k) captures global phenomenon, and rather change relatively slowly over the iterations. We can exploit these findings to avoid the naive approach to draw a sample from (1).

In particular, we compute the document-specific sparse term exactly and for the remainder language model term we borrow idea from (Li et al., 2014). We use a slightly stale distribution for the language model. Then Metropolis Hastings (MH) algorithm allows us to convert the stale sample into a fresh one, provided that we compute ratios between successive states correctly. It is sufficient to run MH for a few number of steps because the stale distribution acting as the proposal is very similar to the target. This is because, as pointed out earlier, the language model term does not change too drastically whenever we resample a single word. The number of words is huge, hence the amount of change per word is concomitantly small. (Only if one could convert stale bread into fresh one, it would solve world's food problem!)

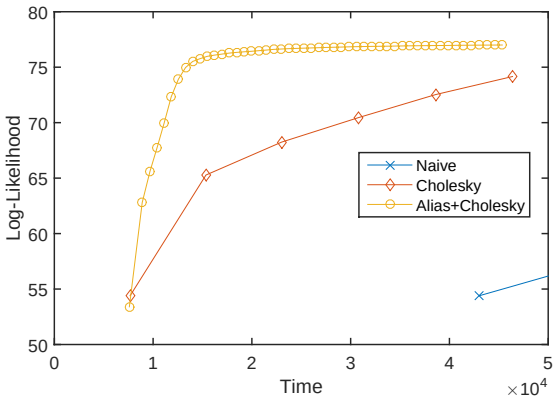

Figure 2: Plot comparing average log-likelihood vs time (in sec) achieved after applying each trick on the NIPS dataset. The shapes on each curve denote end of each iteration.

The exercise of using stale distribution and MH steps is advantageous because sampling from it can be carried out in $O(1)$ amortized time, thanks to alias sampling technique (Vose, 1991). Moreover, the task of building the alias tables can be outsourced to other cores.

With the combination of both Cholesky and Alias tricks, the sampling complexity can thus be brought down to $O(K_d M^2)$ where K_d represents the number of actually instantiated topics in the document and $K_d \ll K$. In particular, we plot the sampling rate achieved naively, with Cholesky (CH) trick and with Cholesky+Alias (A+CH) trick in figure 2 demonstrating better likelihood at much less time. Also after initial few iterations, the time per iteration of A+CH trick is 9.93 times less than CH and 53.1 times less than naive method. This is because initially we start with random initialization of words to topics, but after few iterations the $n_{k,d}$ vector starts to become sparse.

5 Experiments

In this section we evaluate our Word Vector Topic Model on various experimental tasks. Specifically we wish to determine:

- Is our model is able to find coherent and meaningful topics?
- Is our model able to infer the topic distribution of a held-out document even when the document contains words which were previously unseen?

We run our experiments[4] on two datasets 20-NEWSGROUP[5] and NIPS[6]. All the datasets were tokenized and lowercased with `cdec` (Dyer et al., 2010).

5.1 Topic Coherence

Quantitative Analysis Typically topic models are evaluated based on the likelihood of held-out documents. But in this case, it is not correct to compare perplexities with models which do topic modeling on words. Since our topics are continuous distributions, the probability of a word vector is given by its density w.r.t the normal distribution based on its topic assignment, instead of a probability mass from a discrete topic distribution. Moreover, (Chang et al., 2009) showed that higher likelihood of held-out documents doesn't necessarily correspond to human perception of topic coherence. Instead to measure topic coherence we follow (Newman et al., 2009) to compute the Pointwise Mutual Information (PMI) of topic words w.r.t wikipedia articles. We extract the document co-occurrence statistics of topic words from Wikipedia and compute the score of a topic by averaging the score of the top 15 words of the topic. A higher PMI score implies a more coherent topic as it means the topic words usually co-occur in the same document. In the last line of Table 1, we present the PMI score for some of the topics for both Gaussian LDA and traditional multinomial LDA. It can be seen that Gaussian LDA is a clear winner, achieving an average 275% higher score on average.

However, we are using embeddings trained on Wikipedia corpus itself, and the PMI measure is computed from co-occurrence in the Wikipedia corpus. As a result, our model is definitely biased towards producing higher PMI. Nevertheless Wikipedia PMI is a believed to be a good measure of semantic coherence.

Qualitative Analysis Table 1 shows some top words from topics from Gaussian-LDA and LDA on the 20-news dataset for $K = 50$. The words in Gaussian-LDA are ranked based on their density assigned to them by the posterior predictive distribution in the final sample. As shown, Gaussian LDA is able to capture several intuitive topics in the corpus such as sports, government, 'religion', 'universities', 'tech', 'finance' etc. One interesting topic discovered by our model (on both 20-news and NIPS dataset) is the collection of human names, which was not captured by classic LDA. While one might imagine that names associated with particular topics might be preferable to a 'names-in-general' topic, this ultimately is a matter of user preference. More substantively, classic LDA failed to identify the 'finance' topics. We also noticed that there were certain words ('don', 'writes', etc) which often came as a top word in many topics in classic LDA. However our model was not able to capture the 'space' topics which LDA was able to identify.

Also we visualize a part of the continuous space where the word embedding is performed. For this task we performed the Principal Component Analysis (PCA) over all the word vectors and plot the first two components as shown in Figure 3. We can see clear separations between some of the clusters of topics as depicted. The other topics would be separated in other dimensions.

5.2 Performance on document containing new words

In this experiment we evaluate the performance of our model on documents which contains previously unseen words. It should be noted that traditional topic modeling algorithms will typically ignore such words while inferring the topic distribution and hence might miss out important words. The continuous topic distributions of the Word Vector Topic Model on the other hand, will be able

[4] Our implementation is available at `https://github.com/rajarshd/Gaussian_LDA`

[5] A collection of newsgroup documents partitioned into 20 news groups. After pre-processing we had 18768 documents. We randomly selected 2000 documents as our test set. This dataset is publicly available at `http://qwone.com/~jason/20Newsgroups/`

[6] A collection of 1740 papers from the proceedings of Neural Information Processing System. The dataset is available at `http://www.cs.nyu.edu/~roweis/data.html`

Gaussian LDA topics								
hostile	play	government	people	university	hardware	scott	market	gun
murder	round	state	god	program	interface	stevens	buying	rocket
violence	win	group	jews	public	mode	graham	sector	military
victim	players	initiative	israel	law	devices	walker	purchases	force
testifying	games	board	christians	institute	rendering	tom	payments	machine
provoking	goal	legal	christian	high	renderer	russell	purchase	attack
legal	challenge	bill	great	research	user	baker	company	operation
citizens	final	general	jesus	college	computers	barry	owners	enemy
conflict	playing	policy	muslims	center	monitor	adams	paying	fire
victims	hitting	favor	religion	study	static	jones	corporate	flying
rape	match	office	armenian	reading	encryption	joe	limited	defense
laws	ball	political	armenians	technology	emulation	palmer	loans	warning
violent	advance	commission	church	programs	reverse	cooper	credit	soldiers
trial	participants	private	muslim	level	device	robinson	financing	guns
intervention	scores	federal	bible	press	target	smith	fees	operations
0.8302	0.9302	0.4943	2.0306	0.5216	2.3615	2.7660	1.4999	1.1847
Multinomial LDA topics								
turkish	year	people	god	university	window	space	ken	gun
armenian	writes	president	jesus	information	image	nasa	stuff	people
people	game	mr	people	national	color	gov	serve	law
armenians	good	don	bible	research	file	earth	line	guns
armenia	team	money	christian	center	windows	launch	attempt	don
turks	article	government	church	april	program	writes	den	state
turkey	baseball	stephanopoulos	christ	san	display	orbit	due	crime
don	don	time	christians	number	jpeg	moon	peaceful	weapons
greek	games	make	life	year	problem	satellite	article	firearms
soviet	season	clinton	time	conference	screen	article	served	police
time	runs	work	don	washington	bit	shuttle	warrant	control
genocide	players	tax	faith	california	files	lunar	lotsa	writes
government	hit	years	good	page	graphics	henry	occurred	rights
told	time	ll	man	state	gif	data	writes	article
killed	apr	ve	law	states	writes	flight	process	laws
0.3394	0.2036	0.1578	0.7561	0.0039	1.3767	1.5747	-0.0721	0.2443

Table 1: Top words of some topics from Gaussian-LDA and multinomial LDA on 20-newsgroups for $K = 50$. Words in Gaussian LDA are ranked based on density assigned to them by the posterior predictive distribution. The last row for each method indicates the PMI score (w.r.t. Wikipedia co-occurence) of the topics fifteen highest ranked words.

to assign topics to an unseen word, if we have the vector representation of the word. Given the recent development of fast and scalable methods of estimating word embeddings, it is possible to train them on huge text corpora and hence it makes our model a viable alternative for topic inference on documents with new words.

Experimental Setup: Since we want to capture the strength of our model on documents containing unseen words, we select a subset of documents and replace words of those documents by its synonyms if they haven't occurred in the corpus before. We obtain the synonym of a word using two existing resources and hence we create two such datasets. For the first set, we use the Paraphrase Database (Ganitkevitch et al., 2013) to get the lexical paraphrase of a word. The paraphrase database[7] is a semantic lexicon containing around 169 million paraphrase pairs of which 7.6 million are lexical (one word to one word) paraphrases. The dataset comes in varying size ranges starting from S to XXXL in increasing order of size and decreasing order of paraphrasing confidence. For our experiments we selected the L size of the paraphrase database.

The second set was obtained using WordNet (Miller, 1995), a large human annotated lexicon for English that groups words into sets of synonyms called synsets. To obtain the synonym of a word, we first label the words with their part-of-speech using the Stanford POS tagger (Toutanova et al., 2003). Then we use the WordNet database

[7] http://www.cis.upenn.edu/~ccb/ppdb/

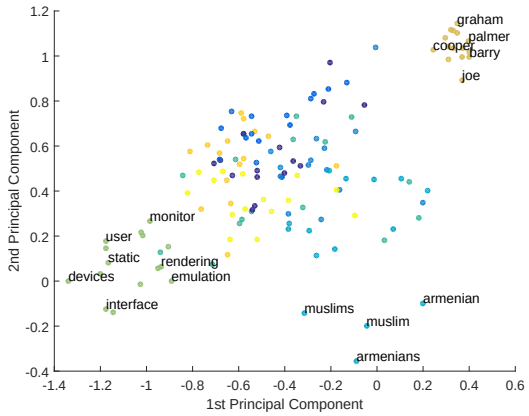

Figure 3: The first two principal components for the word embeddings of the top words of topics shown in Table 1 have been visualized. Each blob represents a word color coded according to its topic in the Table 1.

to get the synonym from its sysnset.[8] We select the first synonym from the synset which hasn't occurred in the corpus before. On the 20-news dataset (vocab size = 18,179 words, test corpus size = 188,694 words), a total of 21,919 words (2,741 distinct words) were replaced by synonyms from PPDB and 38,687 words (2,037 distinct words) were replaced by synonyms from Wordnet.

Evaluation Benchmark: As mentioned before traditional topic model algorithms cannot handle OOV words. So comparing the performance of our document with those models would be unfair. Recently (Zhai and Boyd-Graber, 2013) proposed an extension of LDA (*infvoc*) which can incorporate new words. They have shown better performances in a document classification task which uses the topic distribution of a document as features on the 20-news group dataset as compared to other fixed vocabulary algorithms. Even though, the *infvoc* model can handle OOV words, it will most likely not assign high probability to a new topical word when it encounters it for the first time since it is directly proportional to the number of times the word has been observed On the other hand, our model could assign high probability to the word if its corresponding embedding gets a high probability from one of the topic gaussians. With the experimental setup mentioned before, we want to evaluate performance of this property of our model. Using the topic distribution of a document as features, we try to classify the document into one of the 20 news groups it belongs to. If the document topic distribution is modeled well, then our model should be able to do a better job in the classification task.

To infer the topic distribution of a document we follow the usual strategy of fixing the learnt topics during the training phase and then running Gibbs sampling on the test set (G-LDA (*fix*) in table 2). However *infvoc* is an online algorithm, so it would be unfair to compare our model which observes the entire set of documents during test time. Therefore we implement the online version of our algorithm using Gibbs sampling following (Yao et al., 2009). We input the test documents in batches and do inference on those batches independently also sampling for the topic parameter, along the lines of *infvoc*. The batch size for our experiments are mentioned in parentheses in table 2. We classify using the multi class logistic regression classifier available in `Weka` (Hall et al., 2009).

It is clear from table 2 that we outperform *infvoc* in all settings of our experiments. This implies that even if new documents have significant amount of new words, our model would still do a better job in modeling it. We also conduct an experiment to check the actual difference between the topic distribution of the original and synthetic documents. Let h and h' denote the topic vectors of the original and synthetic documents. Table 3 shows the average l_1, l_2 and l_∞ norm of $(h - h')$ of the test documents in the NIPS dataset. A low value of these metrics indicates higher similarity. As shown in the table, Gaussian LDA performs better here too.

6 Conclusion and Future Work

While word embeddings have been incorporated to produce state-of-the-art results in numerous supervised natural language processing tasks from the word level to document level ; however, they have played a more minor role in unsupervised learning problems. This work shows some of the promise that they hold in this domain. Our model can be extended in a number of potentially useful, but straightforward ways. First, DPMM models of word emissions would better model the fact that identical vectors will be generated multiple times, and perhaps add flexibility to the topic distributions that can be captured, without sacrificing our

[8] We use the `JWI` toolkit (Finlayson, 2014)

Model	Accuracy	
	PPDB	WordNet
infvoc	28.00%	19.30%
G-LDA (*fix*)	44.51%	43.53%
G-LDA (*1*)	44.66%	43.47%
G-LDA (*100*)	43.63%	43.11%
G-LDA (*1932*)	44.72%	42.90%

Table 2: Accuracy of our model and *infvoc* on the synthetic datasets. In Gaussian LDA *fix*, the topic distributions learnt during training were fixed; G-LDA(*1, 100, 1932*) is the online implementation of our model where the documents comes in mini-batches. The number in parenthesis denote the size of the batch. The full size of the test corpus is 1932.

Model	PPDB (Mean Deviation)		
	L_1	L_2	L_∞
infvoc	94.95	7.98	1.72
G-LDA (*fix*)	15.13	1.81	0.66
G-LDA (*1*)	15.71	1.90	0.66
G-LDA (*10*)	15.76	1.97	0.66
G-LDA (*174*)	14.58	1.66	0.66

Table 3: This table shows the Average L_1 Deviation, Average L_2 Deviation, Average L_∞ Deviation for the difference of the topic distribution of the actual document and the synthetic document on the NIPS corpus. Compared to *infvoc*, G-LDA achieves a lower deviation of topic distribution inferred on the synthetic documents with respect to actual document. The full size of the test corpus is 174.

preference for topical coherence. More broadly still, running LDA on documents consisting of different modalities than just text is facilitated by using the *lingua franca* of vector space representations, so we expect numerous interesting applications in this area. An interesting extension to our work would be the ability to handle polysemous words based on multi-prototype vector space models (Neelakantan et al., 2014; Reisinger and Mooney, 2010) and we keep this as an avenue for future research.

Acknowledgments

We thank the anonymous reviewers and Manaal Faruqui for helpful comments and feedback.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of ICML*.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL*.

Mark Finlayson, 2014. *Proceedings of the Seventh Global Wordnet Conference*, chapter Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation, pages 78–85.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, April.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of EMNLP*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641*.

Pengfei Hu, Wenju Liu, Wei Jiang, and Zhanlei Yang. 2012. Latent topic model based on Gaussian-LDA for audio retrieval. In *Pattern Recognition*, volume 321 of *CCIS*, pages 556–563. Springer.

Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*.

David Newman, Sarvnaz Karimi, and Lawrence Cavedon. 2009. External evaluation of topic models. pages 11–18, December.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10.

G. Stewart. 1998. *Matrix Algorithms*. Society for Industrial and Applied Mathematics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proc. of ACL*.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning : Vector space models of semantics. *JAIR*, pages 141–188.

Peter D. Turney. 2006. Similarity of semantic relations. *Comput. Linguist.*, 32(3):379–416, September.

Michael D. Vose. 1991. A linear algorithm for generating random numbers with a given distribution. *Software Engineering, IEEE Transactions on*.

Li Wan, Leo Zhu, and Rob Fergus. 2012. A hybrid neural network-latent topic model. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, volume 22, pages 1287–1294.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA. ACM.

Ke Zhai and Jordan L. Boyd-Graber. 2013. Online latent dirichlet allocation with infinite vocabulary. In *ICML (1)*, volume 28 of *JMLR Proceedings*, pages 561–569. JMLR.org.