Auto-encoding Documents for Topic Modeling with L-2 sparsity regularization

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a novel yet simple neural network architecture for topic modeling. The method is based on training an autoencoder structure where the bottleneck represents the space of the topics distributions and the decoder output represents the space of the word distribution over the topics. We also exploit an auxiliary decoder to prevent mode collapsing in our model. A key feature for an effective topic modeling method is having sparse topic and word distributions, where there is a trade-off between the sparsity level of topics and words. This feature is implemented in our model by an L-2 regularization and the model hyperparameters take care of the trade-off. We show in our experiments that our model achieves competitive results compared to the state-of-the-art deep models for topic modeling, despite its simple architecture and training procedure. The *New York Times* and *20 Newsgroups* datasets are used in the experiments.

1 Introduction

Topic models are among the key models in Natural Language Processing (NLP) that aim to represent a large body of text using only a few concepts or topics, on a completely unsupervised basis. Topic modeling has found its application in many different areas including bioinformatics [13], computer vision [5], recommendation systems [8, 15], etc. Latent semantic indexing (LSI) [4] and probabiliestic latent semantic indexing (PLSI) [6] are among the oldest algorithms, but Latetnt Dirichlet Allocation (LDA) [3] is the most widely used algorithm for topic modeling and most of the successful algorithms in this area are variants of LDA. The main challenge in training the LDA model is its relatively complicated inference model, which makes finding the true posterior a hard task and therefore LDA-based topic modeling algorithms rely on approximating methods. There are many ways to approximate the inference in LDA, e.g. Mean field variational methods [2], variational inference [2], expectation propagation [10], collapsed Gibbs sampling [11], factorization inference [1] etc.

With the advances in deep learning, there has been some efforts to implement LDA and its variants using neural networks. Recent variational autoencoding (VAE) [7, 12] model has paved the way for approximating posteriors using a black box neural network model. However, VAE works best with the posterior approximating distributions for which sampling can be done using reparameterization trick, e.g. Gaussian distribution. Therefore, successful deep models that use VAE framework to implement LDA either directly replace the Dirichlet distribution with a Gaussian distribution [9] or approxiate the Dirichlet using a combination of Gaussians [14]. On the other hand, imposing sparsity constraint, which is an important aspect of LDA and Dirichlet distribution, is not trivial in these models.

In this work we propose a new deep topic modeling algorithm, based on training an autoencoder, which takes as its input the distribution of words in each document and represents the topics in its

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

bottleneck layer. Our model keeps the most important properties of LDA, while having a simple network structure and an easy training procedure. We maintain the sparsity property of the Dirichlet distribution by imposing an L-2 regularization on the softmax layers of the neural network. We also resolve the mode collapsing issue of the model by adding an auxiliary decoder network that makes the separation of the representations in the latent space easier.

2 Model Description

In this section we first briefly describe LDA and its properties and then explain our model and how it resembles the LDA properties without having the difficulties of dealing with the intractable posterior.

2.1 Following LDA properties

Let's assume we have a set of D documents with vocabulary size N. Each document is represented by a vector $\mathbf{x}$. Also there are K topics with different distributions over the words, denoted by $\boldsymbol{\beta}_1$ to $\boldsymbol{\beta}_K$. The LDA generative process is as follows:

For each document $\mathbf{x}$
  Draw topic distribution $\theta \sim$ Dirichlet(α)
  For each word in the document
    Sample a topic $\mathbf{z} \sim$ Multinomial(θ)
    Sample a word in position n, $w_n \sim$ Multinomial($\boldsymbol{\beta}_{\mathbf{z}}$)

Two important objectives that LDA implicitly tries to achieve and they make this model suitable for topic modelling are:

- Distribution of the topics for each document is sparse, therefore each document can be represented by a few topics
- Distribution of the words for each topic is sparse, therefore each topic can be represented by a few words.

There is a trade-off between these two objectives. If a document is represented using only a few topics, then number of words with high probability in those topics should be large, and if topics are represented using only a few words then we need a large number of topics to cover the words in the document. The sparsity of the distributions is a property of the Dirichlet distribution that is controlled by its concentration parameters. Also, based on LDA, the distribution of the words in a document is a mixture of multinomials.

In our model we follow the main principals of the LDA algorithm, i.e. sparse distributions for the topics and words and the final distribution of the words in a document is a mixture of multinomial. On the other hand, we try to avoid the difficulties of training the LDA model. Since our downstream task is finding topics in the documents, and not generating new documents, we do not need to learn the true posterior probability, or find ways to approximate it. Therefore we leave the latent representation unconstrained with regard to its distribution.

We first encode the documents to the topic space $\mathcal{Z}$ using $f_{\text{topic}}(\mathbf{x}; \phi)$, which is implemented by a neural network with parameter set ϕ. To make sure $\mathcal{Z}$ is a probability space we use a softmax layer at the last layer of this network. Also, K vectors, $\boldsymbol{\beta}_1$ to $\boldsymbol{\beta}_K$, with softmax activation represent the words distributions in the topics, each of them is a multinomial distribution. A mixture of multinomials, i.e. $\tilde{\mathbf{x}} = f_\beta(\mathbf{z}) = \sum_{k=1}^{K} z_k \boldsymbol{\beta}_k$, will be a reconstruction of the input vector $\mathbf{x}$. We intentionally do not use a matrix multiplication notation so that we can explain the constraints on $\boldsymbol{\beta}_k$'s in a simpler and more explicit way.

To make both topic and words distributions sparse, we impose an L-2 norm constraint on them. Maximizing the L-2 norm over a positive, sum-to-one vector, concentrates the probability mass over a few number of elements. This way we are keeping the most informative words of each topic,

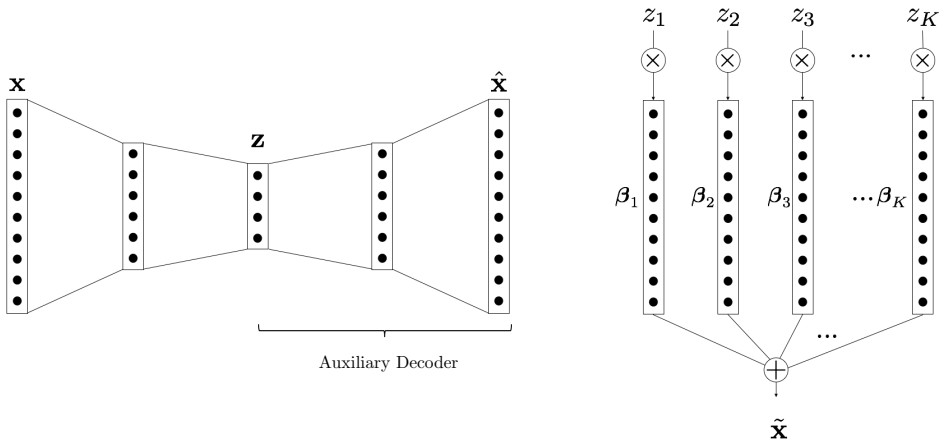

Figure 1: Networks of the model

because sparsity means we are minimizing the entropy of the distributions, Therefore the objective of the algorithm will be as follows:

$$\min_{\phi,\boldsymbol{\beta}_k} \quad D(\mathbf{x},\tilde{\mathbf{x}}) - \gamma \parallel \mathbf{z} \parallel^2 - \eta \sum_{k=1}^{K} \parallel \boldsymbol{\beta}_k \parallel^2, \tag{1}$$

where distance D is the cross entropy, and γ and η are hyperparameters of the model. The trade-off of the sparsity in topics and words distributions can be controlled by tuning γ and η.

2.2 Mode Collapsing

We observed that training the model using Eq. (1) causes mode collapsing, in the sense that only a very few topics will have meaningful words in them and the rest of the topics have high probability over some random words. Also, all the probability mass of the topics distribution for all of the documents are concentrated on those specific topics. In other words, all the documents are encoded to the same set of topics and the model cannot capture the variations in the documents. We believe this is due to the fact that $f_\beta(\mathbf{z})$ is not a powerful function for backpropagating the error signal from the output to the previous layers of the network. To resolve this issue and produce a richer $\mathcal{Z}$ space, we attach an auxiliary decoder to the latent representation, which we call it $f_{\text{AUX}}(\mathbf{z};\varphi)$ and it is a neural network with parameter set φ. The output of this decoder, denoted by $\hat{\mathbf{x}}$, also reconstructs the input document. Our observations show that by adding this decoder we can separate the documents' representations in the latent space.

In both topic and word level, instead of sampling, we consider $\mathbf{z}$ and $\boldsymbol{\beta}_k$'s (for all $k \in \{1,2,..,K\}$) as a normalized typical set of the distribution $\mathbf{z}$ and $\boldsymbol{\beta}_k$'s. This is to avoid sampling from the multinomial distribution for which there is no easy way, e.g. reparameteraztion trick in [7] for Gaussian family, to backpropagate the error for training the neural networks. Therefore the overall objective of our model is:

$$\begin{aligned} \min_{\phi,\varphi,\boldsymbol{\beta}_k} \quad & D(\mathbf{x},\tilde{\mathbf{x}}) + \lambda D(\mathbf{x},\hat{\mathbf{x}}) - \gamma \parallel \mathbf{z} \parallel^2 - \eta \sum_{k=1}^{K} \parallel \boldsymbol{\beta}_k \parallel^2 \\ \text{s.t} \quad & \sum_{n=1}^{N} \beta_{kn} = 1 \qquad \forall k \in \{1,2,...,K\} \\ & \sum_{k=1}^{K} z_k = 1, \end{aligned} \tag{2}$$

where λ is another hyperparameter of the model that controls the role of the auxiliary decoder in training. Figure 1 shows the structure of the networks.

3 Experiments

In this section we compare the performance of the proposed algorithm with LDA with collapsed Gibbs, and two deep models, i.e. Neural Variational Document Model (NVDM) and ProdLDA algorithms in [14, 9]. Although comparing different topic modeling results qualitatively is a hard task, we follow standard metrics for such comparisons. The comparison is made based on *topic coherence* (higher is better) and *perplexity score* (lower is better) of the results.

3.1 New York Times

This dataset consists of $D = 8,447$ documents with vocabulary size $N = 3,012$ words. We downloaded the dataset from this git repository `https://github.com/moorissa/nmf_nyt`. This dataset doesn't need a preprocessing phase, as the common words and stop words has already been removed from it. We try performing topic modeling using 25 and 50 topics for this dataset (CG in the tables mean Collapsed Gibbs and the best results are indicated by bold symbols).

Number of Topics	LDA with CG	ProdLDA	NVDM	Our Model
25	0.26	0.30	0.25	**0.32**
50	0.23	**0.30**	0.21	0.29

Table 1: Topic Coherence for the *New York Times* dataset

Number of Topics	LDA with CG	ProdLDA	NVDM	Our Model
25	781	910	842	**762**
50	770	930	892	**751**

Table 2: Perplexity for the *New York Times* dataset

In this experiment, for $K = 25$ topics the value of hyperparameters are: $\lambda = 0.1$, $\gamma = 0.1$, and $\eta = 0.001$. For $K = 50$ topics these values are: $\lambda = 0.1$, $\gamma = 0.05$, and $\eta = 0.001$.

3.2 20 Newsgroups

The 20 Newsgroups has $D = 11,000$ training documents. We follow the same preprocessing in [14], tokenization, removing some of the non UTF-8 characters and English stop word removal. These are all done using `scikit-learn` package. After this preprocessing the vocabulary size is $N = 2,000$. For this dataset we try training the models with 50 and 200 topics.

Number of Topics	LDA with CG	ProdLDA	NVDM	Our Model
50	0.18	0.23	0.10	**0.25**
200	0.14	**0.19**	0.08	0.18

Table 3: Topic Coherence for the *20 Newsgroups* dataset

Number of Topics	LDA with CG	ProdLDA	NVDM	Our Model
50	**737**	1180	830	795
200	**690**	1139	842	806

Table 4: Perplexity for the *20 Newsgroups* dataset

In this experiment, for $K = 50$ topics the value of hyperparameters are: $\lambda = 0.5$, $\gamma = 0.1$, and $\eta = 0.001$. For $K = 200$ topics these values are: $\lambda = 0.4$, $\gamma = 0.01$, and $\eta = 0.001$.

We can see that for both datasets, our algorithm achieves competetive results with some of the state-of-the-art deep models for topic modeling. ProdLDA shows better performance in term of the topic coherence when the number of topics gets large. However, for lower number of topics our model outperforms all othe algorithms. We can also see that our results is better than ProdLDA in terms of perplexity for both datasets, although LDA with collapsed Gibbs yilelds the best results for the *20 Newsgroups* dataset. Some random topics and the 10 highest probable words in them are shows in tables 5 and 6 in appendix.

References

[1] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, pages 280–288, 2013.

[2] D. M. Blei, M. I. Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[6] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[7] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of ICLR*, 2014.

[8] X. Li and J. She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 305–314. ACM, 2017.

[9] Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736, 2016.

[10] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[11] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[12] D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

[13] S. Rogers, M. Girolami, C. Campbell, and R. Breitling. The latent process decomposition of cdna microarray data sets. *IEEE/ACM transactions on computational biology and bioinformatics*, 2(2):143–156, 2005.

[14] A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[15] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM, 2015.

A Some results

Here we present some randomly selected topics with their top 10 words with the highest probabilities for the two datasets.

New York Times

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
building	police	percent	preseident	military
city	man	price	executive	plane
build	kill	market	chief	flight
open	arrest	rate	director	fly
house	officer	rise	name	mission
project	pfficial	fall	vice	attack
site	fire	sale	chairman	airline
street	charge	share	company	security
space	shoot	report	agency	air
home	death	increase	management	pilot

Table 5: Topc 10 words in randomly selected topics

20 Newsgroups

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
windows	jews	goverment	space	game
card	israel	people	station	team
drive	israeli	federal	moon	games
monitor	law	american	orbit	play
pc	anti	political	launch	win
drivers	jewish	civil	nasa	season
disk	arab	war	shuttle	players
mouse	religious	society	developed	hit
mac	killed	national	object	baseball
printer	muslim	majority	lunar	league

Table 6: Topc 10 words in randomly selected topics