[Paper review 40]

Topic Modeling in Embedding Spaces

(Dieng , et al., 2019)

[Contents]

1. Abstract
2. Introduction
3. Related Works
4. Background
   1. LDA
   2. Word Embedding
5. The Embedded Topic Model
6. Inference and Estimation
7. Conclusion

1. Abstract

Existing topic models : hard to solve with **LARGE VOCABULARIES**!

$\rightarrow$ This paper suggests **"ETM (Embedded topic model)"**

"ETM (Embedded topic model)"

- generative model of documents
- combines **TOPIC MODEL + WORD EMBEDDING**
- inner product between
  - (1) **word embedding**
  - (2) **embedding of its assigned topic**
- to fit ETM, develop **"efficient amortized VI"**

2. Introduction

Topic Modeling

Most topic models are built on LDA

- can be fit to large datasets, using VI & Stochastic Optimization
- but fails in large vocabularies

(need pruning, thus remove important terms !)

Word Embedding

- one-hot encoding : sparse(distributed)representation

  word embedding : dense representation
- crucial in NLP

ETM = **(1) Topic Modeling** + **(2) Word Embedding**

- Topic Model
  - discovers interpretable latent semantic structure of texts
- Word Embedding
  - provides low-dim representation
  - thus, robust to stop words

ETM vs LDA

(sim)

- generative probabilistic model

  (1. each document = mixture of topics)

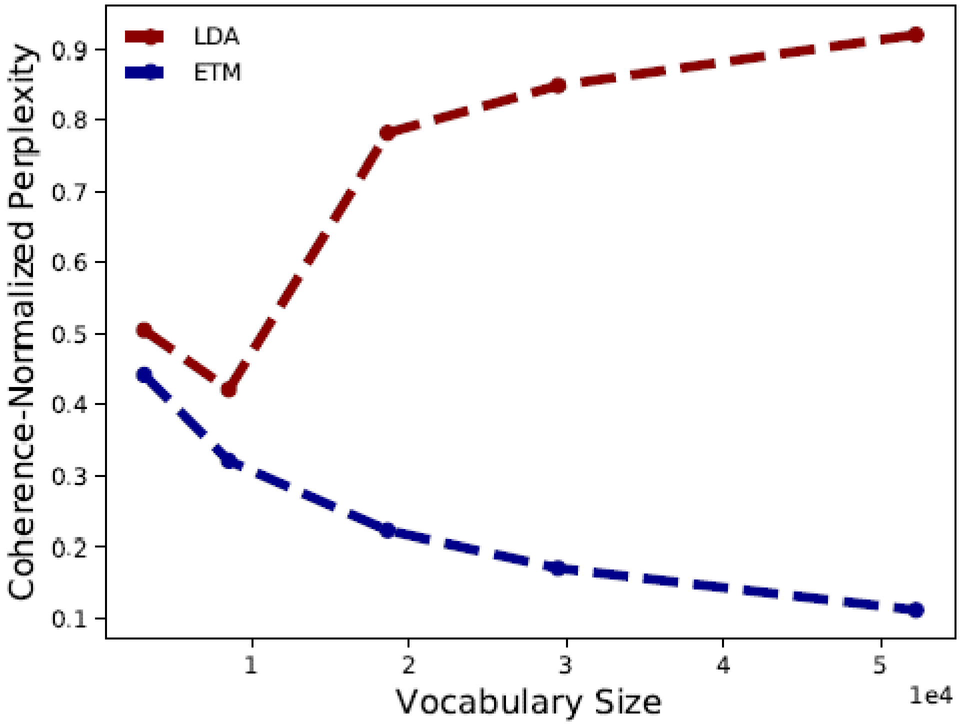
  (2. words = assigned to topics)

(dissim)

- per-topic conditional probability of a term = **"log-linear" form**
- each term = "embedded"
- (lots of topic models have) intractable posterior

  ETM uses VI & amortized inference to approximate topic proportions
- can fit to large vocabs!

Much better performance than LDA!



3. Related Works

Develops a new topic model that extends LDA

one goal of ETM : **"incorporating WORD SIMILARITY into the topic model"**

Previous works with the goal above

- modify topic priors
- modify topic assignment priors
- extend LDA to directly involve word embedding

  (topics = Gaussian with latent mean & cov)

  (likelihood over the embeddings = Gaussian)

How does ETM differ from above?

"It is a model of categorical data, one that goes through the **embedding matrix**"

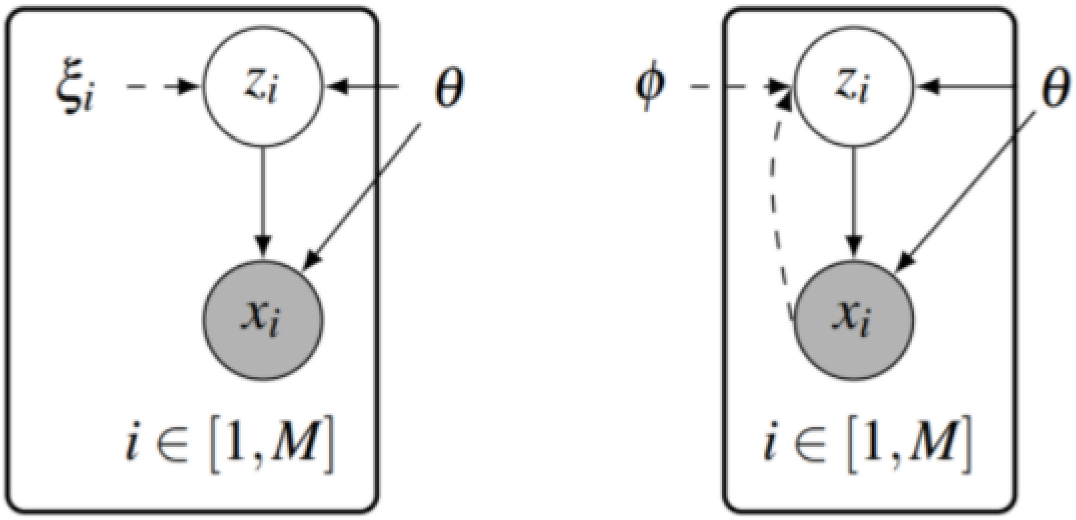
(thus, do not require pre-fitted embeddings & can learn embeddings)

Previous works to combine LDA + Embeddings

- ex) using DNN

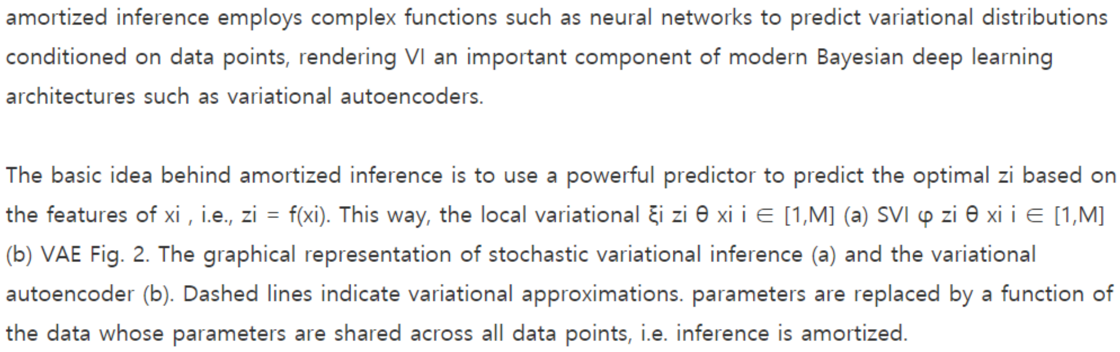
  $\rightarrow$ reduce dimension of text through amortized inference & VAE
- ETM also uses amortized inference methods!

Amortized Inference?



(a) SVI (b) VAE

- (a) Standard VI
- (b) using Amortized Inference



amortized inference employs complex functions such as neural networks to predict variational distributions conditioned on data points, rendering VI an important component of modern Bayesian deep learning architectures such as variational autoencoders.

The basic idea behind amortized inference is to use a powerful predictor to predict the optimal zi based on the features of xi , i.e., zi = f(xi). This way, the local variational ξi zi θ xi i ∈ [1,M] (a) SVI φ zi θ xi i ∈ [1,M] (b) VAE Fig. 2. The graphical representation of stochastic variational inference (a) and the variational autoencoder (b). Dashed lines indicate variational approximations. parameters are replaced by a function of the data whose parameters are shared across all data points, i.e. inference is amortized.

(https://arxiv.org/pdf/1711.05597)

4. Background

ETM = (1) LDA + (2) word embedding

notation :

- corpus of D documents
- V vocabularies
- $w_{dn} \in \{1, \ldots, V\}$: n^{th} word in the d^{th} document

  LDA is a probabilistic generative model of documents (Blei et al., 2003). It posits K topics $\beta_{1:K}$, each of which is

4-1. LDA

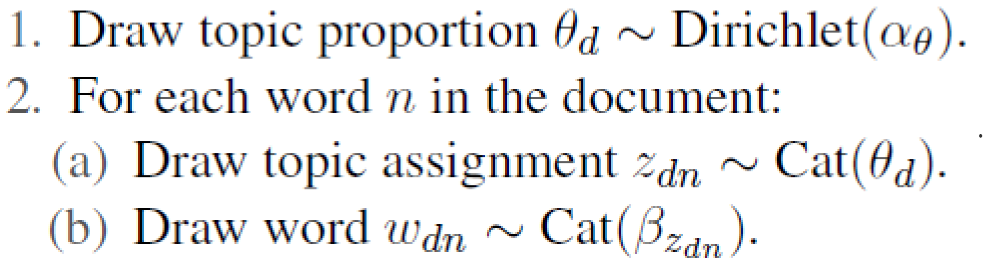
probabilistic generative model of documents

K topics (= $\beta_{1:K}$)

- each are **distribution over vocab**

(ex : topic 1... β_1 : (0.01, 0.03, ..., 0.003) with dimension V)

Algorithm



```
1. Draw topic proportion $\theta_d \sim \text{Dirichlet}(\alpha_\theta)$.
2. For each word $n$ in the document:
    (a) Draw topic assignment $z_{dn} \sim \text{Cat}(\theta_d)$.
    (b) Draw word $w_{dn} \sim \text{Cat}(\beta_{z_{dn}})$.
```

4-2. Word Embedding

Vector representations of words!

ex) word2vec (CBOW & Skip gram)

ETM focuses on **CBOW** (continuous bag-of-words)

likelihood of CBOW : $w_{dn} \sim \operatorname{softmax}(\rho^\top \alpha_{dn})$.

- ρ : embedding matrix (dim = $L \times V$)
    - L : reduced dimension
    - V : number of vocab
    - ρ_v : embedding of vocabulary v (= v^{th} column)
- α_{dn} : context embedding

  (sum of the context embedding vectors (α_v for each word v) of the words surrounding w_{dn})

5. The Embedded Topic Model

embeds both WORDS & TOPICS (contains 2 notions of latent dimension)

- 1) embed vocab in L dim space
- 2) represent each document in terms of K latent topics
    - traditional) each topic = distribution over the vocab
    - ETM) k^{th} topic = vector $\alpha_k \in \mathbb{R}^L$.

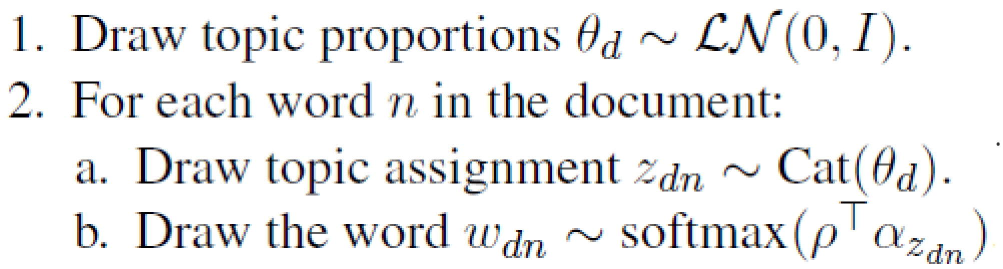
      (α_k = **"topic embedding"**)

ETM uses **log-linear model** that takes inner product of..

- **1) word-embedding matrix**
- **2) topic-embedding**

$\rightarrow$ high probability to a word v in topic k, by measuring the **"agreement between the word's embedding & topic's embedding "**

Generative Process:



1. Draw topic proportions $\theta_d \sim \mathcal{LN}(0, I)$.
2. For each word n in the document:
   a. Draw topic assignment $z_{dn} \sim \mathrm{Cat}(\theta_d)$.
   b. Draw the word $w_{dn} \sim \mathrm{softmax}(\rho^\top \alpha_{z_{dn}})$

Other steps are same as the traiditonal ones, except...

(Step 1) Logistic Normal distribution

- transforms standard Gausian r.v. to the simplex
- $\delta_d \sim \mathcal{N}(0, I); \quad \theta_d = \mathrm{softmax}(\delta_d).$
- Why not Dirichlet?

  $\rightarrow$ for **"reparameterization trick"**

(Step 2-b) Softmax of inner product between..

- 1) embeddings of the vocab ρ
- 2) assigned topic embedding $\alpha_{z_{dn}}$
- looks like CBOW likelihood! (= $w_{dn} \sim \mathrm{softmax}(\rho^\top \alpha_{dn})$)
  - CBOW) uses "surrounding words" to form context vector α_{dn}
  - ETM) uses topic embedding $\alpha_{z_{dn}}$ as context vector

Two ways of using embeddings

- method 1) previously fitted embeddings
  - learns the topics of a corpus in a particular embeddings
- method 2) learning them as a part of fitting
  - simultaneously find topics & embeddings

6. Inference and Estimation

corpus of documents : $\{\mathrm{w}_1, \ldots, \mathrm{w}_D\}$

parameters of ETM

- embeddings : $\rho_{1:V}$
- topic embeddings : α_k

Marginal Likelihood

maximize marginal likelihood of documents!

$\mathcal{L}(\alpha, \rho)=\sum_{d=1}^{D} \log p\left(\mathbf{w}_{d} \mid \alpha, \rho\right)$...all the document

- $p\left(\mathbf{w}_{d} \mid \alpha, \rho\right)=\int p\left(\delta_{d}\right) \prod_{n=1}^{N_{d}} p\left(w_{d n} \mid \delta_{d}, \alpha, \rho\right) \mathrm{d} \delta_{d}$..............................one document
  - intractable!
  - $p\left(w_{d n} \mid \delta_{d}, \alpha, \rho\right)=\sum_{k=1}^{K} \theta_{d k} \beta_{k, w_{d n}}$ .... one word
    - $\theta_{d k}$: topic proportions
    - $\beta_{k v}$: traditional topics ($\beta_{k v}=\operatorname{softmax}\left(\rho^{\top} \alpha_{k}\right)\Big|_{v}$)

      (= distribution over words)

Variational Inference

optimize a sum of per-document bounds on the log of the marginal likelihood

(= $p\left(\mathbf{w}_{d} \mid \alpha, \rho\right)=\int p\left(\delta_{d}\right) \prod_{n=1}^{N_{d}} p\left(w_{d n} \mid \delta_{d}, \alpha, \rho\right) \mathrm{d} \delta_{d}$)

two parameters to optimize

- model params (= embedding & topic embedding)
- variational params

Amortized inference $q\left(\delta_{d} ; \mathbf{w}_{d}, \nu\right)$

- δ_{d} depends on both $\mathbf{w}_{d}$ and shared variational params ν
- Gaussian (whose mean & var from NN (inference network))

  (get document $\mathbf{w_d}$ as input, outputs mean & var of δ_d)

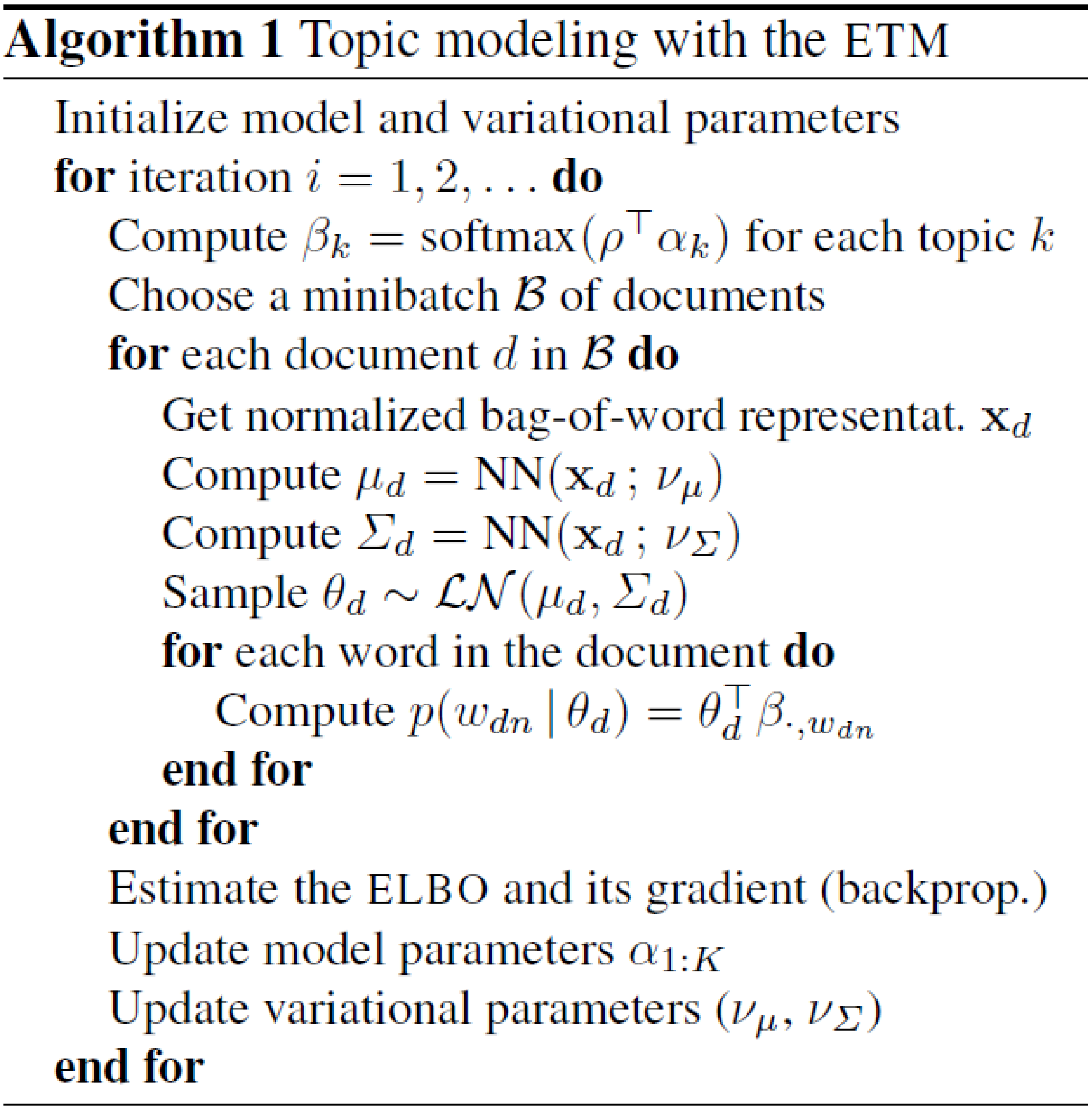
ELBO : $\mathcal{L}(\alpha, \rho, \nu)=\sum_{d=1}^{D} \sum_{n=1}^{N_{d}} \mathbb{E}_{q}\left[\log p\left(w_{n d} \mid \delta_{d}, \rho, \alpha\right)\right]-\sum_{d=1}^{D} \mathrm{KL}\left(q\left(\delta_{d} ; \mathbf{w}_{d}, \nu\right) \| p\left(\delta_{d}\right)\right)$

- 1st term) encourage to place mass on topic proportions δ_d (that explain the observed word)
- 2nd term) stay close to prior $p(\delta_d)$ (regularizer)

Techniques used..

- MC approximation
- reparam trick
- data subsampling
- Adam optimizer

Topic modeling with ETM



Algorithm 1 Topic modeling with the ETM

```
Initialize model and variational parameters
for iteration $i = 1, 2, \ldots$ do
    Compute $\beta_k = \mathrm{softmax}(\rho^\top \alpha_k)$ for each topic $k$
    Choose a minibatch $\mathcal{B}$ of documents
    for each document $d$ in $\mathcal{B}$ do
        Get normalized bag-of-word representat. $\mathbf{x}_d$
        Compute $\mu_d = \mathrm{NN}(\mathbf{x}_d\,;\nu_\mu)$
        Compute $\Sigma_d = \mathrm{NN}(\mathbf{x}_d\,;\nu_\Sigma)$
        Sample $\theta_d \sim \mathcal{LN}(\mu_d, \Sigma_d)$
        for each word in the document do
            Compute $p(w_{dn}\,|\,\theta_d) = \theta_d^\top \beta_{\cdot,w_{dn}}$
        end for
    end for
    Estimate the ELBO and its gradient (backprop.)
    Update model parameters $\alpha_{1:K}$
    Update variational parameters $(\nu_\mu, \nu_\Sigma)$
end for
```

.

7. Conclusion

ETM

- generative model of documents that (1) LDA + (2) word embedding
- assumes topics & words live in sambe embedding space
- works well in large-vocab