NEURAL TOPIC MODEL VIA OPTIMAL TRANSPORT

- Review by *Garde Joël*
- Original paper

Neural Topic Model

A *Neural Topic Model* (NTM) is a document analysis model. A document x is represented as a *bag-of-words* $\Delta \in \mathbb{R}^V$ where V is the size of the vocabulary. The aim is to uncover the latent variables $z \in \mathbb{R}^K$, such that $p_\phi(x|z) = \text{Multi}(\phi(z))$ and K is the number of *topics*.

In this model, a *topic* defines a distribution over words (here given by $\phi(z)$) and a document $\Delta \sim \text{Multi}(\phi(z))$.

Optimal Transport

Instead of minimizing the Kullback-Leibler divergence like done in *VAE* framework of NTM, the authors propose to directly measure the distance between the distribution of words in a document $\tilde{x}$ (support $[0,1]^V$) in and the corresponding distribution of topics z (support in $[0,1]^K$).

Pre-learnt embeddings

A significant contribution is the use of pre-trained embeddings (from *Glove*) in the NTM framework. Pre-trained word embeddings are used to defined the cost function c.

Architecture

Notations.

- B: batch size.
- K: number of topics.
- V: vocabulary size.
- L: dimensions of the words embeddings.
- E: gloves embeddings of the vocabulary.
- G: learnt topics embeddings.
- X: document count matrix.

Green-red blocks contain learnt parameters.

Overview.

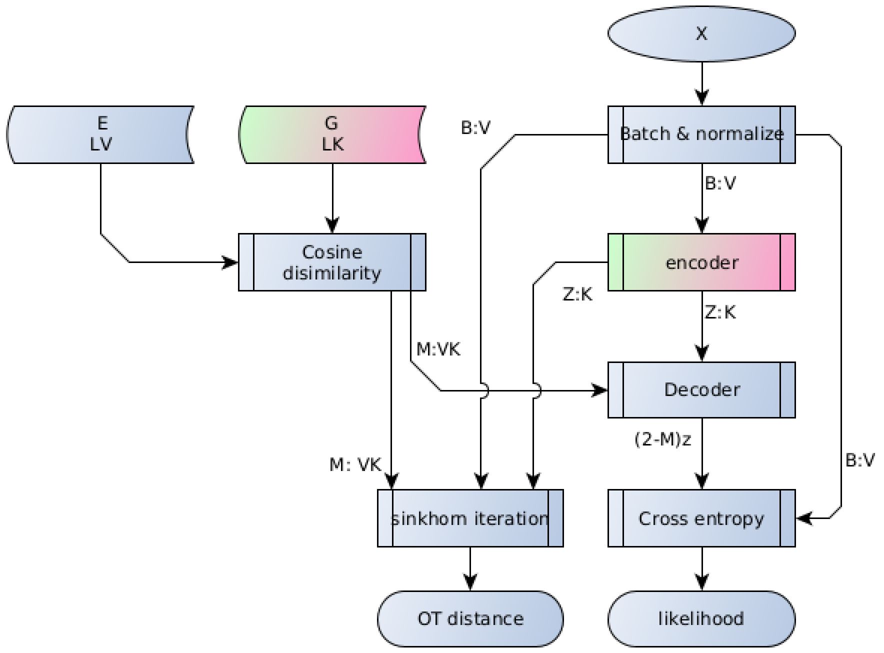

Figure 1: OTNTM architecture

The batch of documents b is normalized to have a total mass of 1. The encoder θ is a neural net with one hidden layer, followed by a softmax. At its output is z, the distribution of topics.

Using the words-embeddings and the learnt topics embeddings, the positive cost matrix c of the transport is computed as $1 - \cos(E^T G)$. The cosine similarity is often used in language processing. Here, the cost is therefore the cosine dis-similarity.

Using c, b and z, the OT distance is approximated by the entropic regularized OT problem, and solved using sinkhorn iterations.

The cross-entropy with the decoded $\hat{x}$ is also computed. The decoder is deterministic and uses the cosine similary between words and topics.

Even if the architecture is generative, the main interest is in learning G, the topics vectors.

Comments

Comparing to generative models.

If the architecture can look similar, the use of OT is quite different that from OT-GAN. Indeed, in NTMOT, OT is computed between the input b and the latent variables z directly. And most importantly, it is not using the *mini-batch sampling loss*. In OTNTM, for each document, its distribution of words is compared to the distribution of topics. OT-GAN, compares the full distribution defined by the input dataset to the generative one, using mini-batch as an approximation. In a sense, the loss in OTNTM is *point-wise*, for each document, while the loss in OT-GAN is *batch-wise*.

Architecture and optimal transport.

This architecture makes good use of optimal transport. b and z do not have the same support, and the cost c uses domain specific information: the glove embeddings.

However, the authors provide very little theoretical explanation as to why there loss is computed as ot distance + ϵ cross-entropy. The strange value of the hyper-parameter ϵ, 0.007, makes this sum even more doubtful.

I believe two improvements could be made to make the architecture cleaner:

- Use the Sinkhorn Divergence SD instead of the entropic-regularized OT cost summed to the cross-entropy.
- Use the unbalanced formulation of the Sinkhorn algorithm instead of manually normalizing b and passing z through a softmax.

This way, it would really make full use of optimal transport.

The architecture is not as scalable as advertised by the authors. It need to instantiate and update at each iteration the cost matrix $c \in \mathbb{R}^V \times \mathbb{R}^K$. Yet in natural language processing application, V is often too large ($\sim 10^5$) to be practical. A more careful approach would restrict c to V_b, the vocabulary restricted to the batch. (They experiment only on specific datasets with a rather small vocabulary.)

Experiment

I run the OTNTM on the Reuters-21578 dataset and perform a qualitative analysis of the resulting topics embeddings.

Using a hand-made Sinkhorn algorithm resulted in `NaN` values. I used the version from OTT to benefit from the `lse_mode` and the handling of zero weights. (There are a lot of zero-weights, since a document contains only a small portion of the total vocabulary).

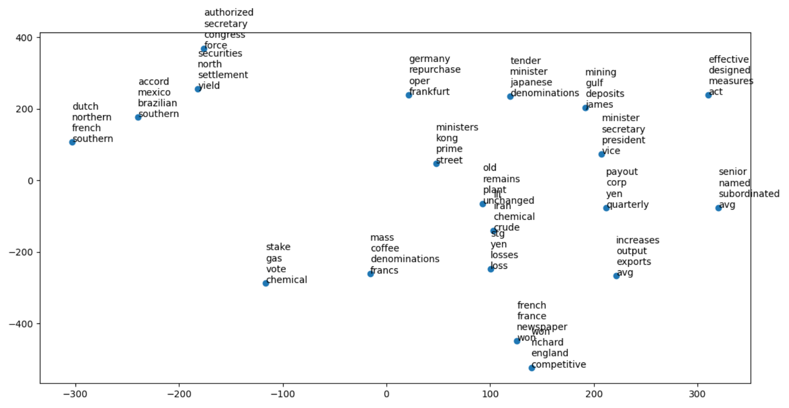

Figure 2: TSNE of 20 topics on reuters-21758

The quality of the topic is not the best, and some are hard to interpret. This is most likely due to having only 20 topics to reduce the computational burden, when the datasets lists around 120 tags for its documents. Nonetheless, we can see some coherent clusters: “france, french, newspaper”, “germany, repurchase oper frankfurt”, “minister secretary president vice”, “stake gas vote chemical”.