Autoencoding variational Bayes for latent Dirichlet allocation

Zach Wolpe[1] and Alta de Waal[12]

[1] Department of Statistics, University of Pretoria
[2] Center for Artificial Intelligence Research (CAIR)

Abstract. Many posterior distributions take intractable forms and thus require variational inference where analytical solutions cannot be found. Variational Inference and Monte Carlo Markov Chains (MCMC) are established mechanism to approximate these intractable values. An alternative approach to sampling and optimisation for approximation is a direct mapping between the data and posterior distribution. This is made possible by recent advances in deep learning methods. Latent Dirichlet Allocation (LDA) is a model which offers an intractable posterior of this nature. In LDA latent topics are learnt over unlabelled documents to soft cluster the documents. This paper assesses the viability of learning latent topics leveraging an autoencoder (in the form of Autoencoding variational Bayes) and compares the mimicked posterior distributions to that achieved by VI. After conducting various experiments the proposed AEVB delivers inadequate performance. Under Utopian conditions comparable conclusion are achieved which are generally unattainable. Further, model specification becomes increasingly complex and deeply circumstantially dependant - which is in itself not a deterrent but does warrant consideration. In a recent study, these concerns were highlighted and discussed theoretically. We confirm the argument empirically by dissecting the autoencoder's iterative process. In investigating the autoencoder, we see performance degrade as models grow in dimensionality. Visualization of the autoencoder reveals a bias towards the initial randomised topics.

Keywords: Autoencoders · Variational Inference · Latent Dirichlet Allocation · Natural Language Processing · Deep Learning .

1 Introduction

High dimensional data such as text, speech, images and spatiotemporal data are typically labelled as big data, not only because of high volumes, but also because of veracity and velocity. It is for these reasons that unsupervised representations are becoming more in demand in order to project the data onto a lower dimensional space that is more manageable. Most often, this involves the computation of a posterior distribution which comes at a high computational expense. One such method is topic modelling which infers latent semantic representations of text. The high dimensional integrals of the posterior predictive posterior distribution of a topic model is intractable and approximation techniques such as sampling (Markov Chain Monte Carlo) or optimization (variational inference) are standard approaches to approximate these integrals. MCMC samples from the proportionate posterior and is guaranteed to converge to the true posterior given enough data and computational time [12]. However, the associated computational costs associated with MCMC makes it impractical for large and high dimensional corpora. On the other hand, variational inference simplifies the estimation procedure by approximating the posterior with a solvable solution [2], but are known for underestimating the posterior variance. Furthermore, for any new topic model with slightly different assumptions, the inference updates for both these techniques need to be derived theoretically.

An alternative approach to sampling and optimization, is to directly map input data to an approximate posterior distribution. This is called an inference network and was introduced by Dayan et al.[4] in 1995. An autoencoding variational Bayes (AEVB) algorithm, or variational autoencoder, trains an inference network [14] to perform this mapping and thereby mimicking the effect of probabilistic inference [15]. Using Automatic Differentiation Variational Inference (ADVI) [7] in combination with AEVB, posterior inference can be performed on almost any continuous latent variable model.

In this paper we describe and investigate the implementation of an autencoder variational Bayes (AEVB) for LDA. We are specifically interested in the quality of posterior distributions it produces. Related work [15] has indicated that a straightforward AEVB implementation does not produce meaningful topics. The two main challenges stated by the authors are the fact that the Dirichlet prior is not a location scale family, and thereby making the reparameterisation problematic. Secondly, because of component collapsing, the inference network becomes stuck in a bad local optimum in which all the topics are indentical. Although Srivastave & Sutton [15] provided a this explanation as well as produced a solution to the problem, our aim is to take a step back and analyse the behaviour of the AEVB on topic models empirically. Our experiments confirm the issues raised by [15] and based on that, we dissect the autoencoder's iterative process in order to understand how and when the autoencoder allocates documents to topics.

The structure of the paper is as follows: In Section 2 we provide background theory on LDA and in section 3 we introduce AEVB in LDA before defining the experiments in Section 4. Section 5 is a dedicated discussion into the the AEVB's performance which is followed by conclusions in section 6.

2 Latent Dirichlet Allocation

LDA is probably the most popular topic model. In LDA, each document is probabilistically assigned to each topic based on the correlation between the words in each document. The generative process of the LDA is as follows[2]: Assuming a corpus consists of K topics, LDA assumes each document is generated by:

1. Randomly choose K topic distributions $\beta_k \sim \text{Dirichlet}(\lambda_\beta)$ over the available dictionary - where β denotes the *topic* × *word* matrix where the probability of the i^{th} word belonging to the j^{th} topic is $\beta_{i,j}$ in β.
2. For each document $d = \{w_1, w_2, .., w_n\}$:
   (a) Randomly choose θ_d, the distribution over topics: a *document* × *topic* matrix.
   (b) For each word w_i randomly select a topic $z_n \sim \text{Multinomial}(\theta_d)$; and within that topic, sample a word $w_n \sim \text{Multinomial}(\beta_{z_n})$.

Figure 1 illustrates the generative model graphically, with plates representing iterations over documents $1, ..., M$ and words $1, ..., N$. The shaded node w is the only observable variable in the model. α is simply a model hyperparameter.

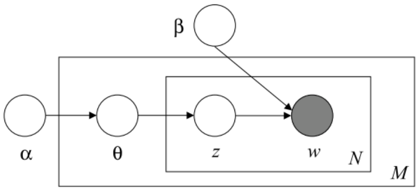

Fig. 1. LDA graphical model.

Under this generative model, the marginal likelihood of a document **w** is [15]:

$$p(\mathbf{w}|\alpha, \beta) = \int_\theta \left(\prod_{n=1}^{N} \sum_{z_n=1}^{k} p(w_n|z_n, \beta) p(z_n|\theta) \right) p(\theta|\alpha) d\theta. \quad (1)$$

Due to the coupling of θ and α under the multinomial assumption, posterior inference over the hidden variables θ and z is intractable.

2.1 Mean field variational inference

As mentioned before, MCMC can be used to approximate the posterior distributions. For the scope of this paper, we focus on optimization techniques. Mean field variational inference (VI) breaks the coupling between θ and z by introducing the free variational parameters γ (over θ) and ϕ (over z). The variational posterior which best approximate the true posterior when optimized is now $q(\theta, z|\gamma, \phi) = q_\gamma(\theta)\prod_n q_\phi(z_n)$, and the optimization problem is to maximize the evidence lower bound (ELBO) [6]

$$L(\gamma, \phi|\alpha, \beta) = D_{KL}[q(\theta, z|\gamma, \phi)||p(\theta, z|\mathbf{w}, \alpha, \beta)] - \log p(\mathbf{w}|\alpha, \beta). \tag{2}$$

In the above equation, D_{KL} is the Kullback-Leibler divergence and is utilized to minimize the distance between the variational and posterior distribution[9]. For LDA, the ELBO has closed form updates due to the conjugacy between the Dirichlet and multinomial distributions [15]. Deriving these closed form updates when there is a need for even slight deviations in assumptions can be cumbersome and impractical. One example is where the practitioner wants to investigate the Poisson instead of the multinomial as a count distribution. One can imagine the far-reaching implications of such a deviation on the coordinate descent equations. AEVB is a method that shows promise to sidestep this issue.

3 AEVB

Before we introduce AEVB, we first need to define autoencoding in general. We use Figure 2 as a simple illustration. An autoencoder is a particular variant of neural network; different in that the input matrix X is mapped to itself $\hat{X}$ as apposed to a response variable Y [12]. Clearly the response is not of interest as - at best - it is a replication of the independent variable. The autoencoder's purpose is rather to examine the hidden layers [5]. If the hidden layers, represented at a lower dimensional space, are able to replicate the input variables we have essentially encoded the same information in a lower dimensional domain. Autoencoders are frequently used to generate data, as random numbers can be fed to this lower dimensional encoding's weights and biases to generate approximate output $\hat{X}$ that is similar to the training data. For probabilistic models with latent variables - as in the case of LDA - it is used to infer the variational parameters of an approximate posterior.

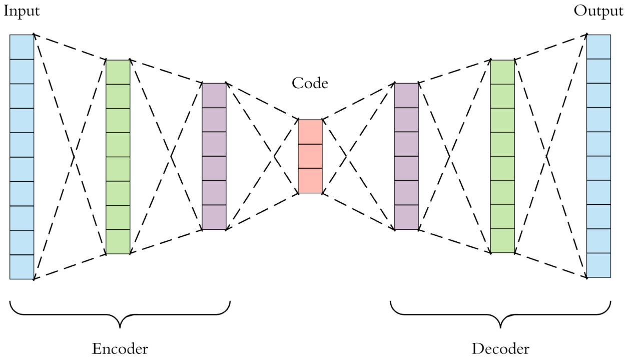

Fig. 2. Autoencoder illustration

3.1 What makes AEVB autoencoding?

From a coding perspective, the latent variables $\mathbf{z}$ can be interpreted as *code*. The variational posterior can be interpreted as a probabilistic *encoder* and the original posterior ($p(\theta, z|\mathbf{w}, \alpha, \beta)$) as a probabilistic *decoder* [6]. The first step in defining the AEVB is to rewrite the ELBO in Eq 2 as [6]:

$$L(\gamma, \phi|\alpha, \beta) = -D_{KL}[q(\theta, z|\gamma, \phi)||p(\theta, z|\mathbf{w}, \alpha, \beta)] + \mathbb{E}_{q(\theta,z|\gamma,\phi)}[\log p(\mathbf{w}|z, \theta, \alpha, \beta)].$$

The first term attempts to match the variational posterior over latent variables to the prior on the latent variables [15]. The second term is crucial in the definition of the AEVB as it ensures that the variational posterior favours values of the latent variables that are good at explaining the data. This can be thought of the reconstruction (or decoder) term in the autoencoder.

3.2 Stochastic Gradient Descent Estimator

Stochastic gradient descent (SGD) - a scalable variation of regular gradient descent - is the optimization algorithm used to minimize the KL divergence (maximize the ELBO) - stochastic in that it computes an approximate gradient as apposed to a true gradient (from a random sample) to speed computation. After initializing the parameters of interest, gradient descent optimizes a specified loss function by iteratively computing the gradient w.r.t each parameter; multiplying the gradient with the learning rate and subtracting the computed quantity from the gradient, formally:

```
Algorithm 1: Stochastic Gradient Descent (SGD)
Input: Training data $S$, learning rate $\eta$, initialization $\sigma$
Output: Model parameters $\Theta = (\gamma, \phi)$
$\gamma \leftarrow 0$; $\phi \leftarrow 0$;
repeat
    for $(x, y) \in S$ do
        $\theta \leftarrow \theta - \eta(\frac{\partial}{\partial\theta} L(\gamma, \phi|\alpha, \beta)$;
    end
until convergence;
```

The learning rate is normally dynamically adjusted to improve efficiency further. The true gradient can be smoothed by adding regularization term to improve ease of computation.

3.3 AEVB for LDA

Autoencoder Variational Bayes (AEVB) is based on ideas from Variational Inference (VI) to offer a potentially scalable alternative. VI works by maximizing the ELBO (Eq. 2) where $q(\theta, z|\gamma, \phi)$ can be thought of as the latent 'code' that describes a fixed x - thus should map the input x the lower-dimensional latent space. $q(\theta, z|\gamma, \phi)$ - the encoder.

Optimizing the objective function tries to map the input variable x to a specified latent space and then back to replicated the input. This structure is indicative of an autoencoder - where the name AEVB comes from. ADVI is used to efficient maximize the ELBO - equivalent to minimizing the original KL divergence. ADVI utilizes Stochastic Gradient Descent, but derivatives are not computed numerically (in the traditional sense), nor symbolically, but rather relies on a representation of variables known as dual numbers to efficiently compute gradients (the details of which are superfluous to this discussion).

But how do we parameterize $q(\theta, z|\gamma, \phi)$ and $p(\theta, z|\mathbf{w}, \alpha, \beta)$? We ought to choose $q(\theta, z|\gamma, \phi)$ such that it can approximate the true posterior $p(\theta, z|\gamma, \phi)$, and $p(\theta, z|\mathbf{w}, \alpha, \beta)$ such that it is flexible enough to represent a vast variety of distributions? Parameterizing these functions with neural networks allows for great flexibility and are efficiently optimized over large datasets.

$q(\theta, z|\gamma, \phi)$ - the encoder - is parameterized such that the code's dimensionality corresponds to a mean and variance of each topic. The parameter space of the decoder is specified as the reciprocal of the encoder. In the case of LDA, the weights and biases of the encoder are specified as:

$$\underset{w\times h}{W_0} \qquad \underset{h\times 1}{b_0} \qquad \underset{h\times 2*(k-1)}{W_1} \qquad \underset{2*(k-1)\times 1}{b_1}$$

where w is the number of words, h the number of hidden layers and k the number of topics. A uniform Dirichlet prior with $\alpha = k$ (number of topics) is specified. The objective (ELBO) is then maximized by ADVI and model parameters are learnt.

4 Experiments

A number of experiments were conducted in aid of answering the follow research questions:

1. *Does the AEVB LDA provide comparable results - and predictive capability - to the well establish - VI learnt - LDA?*
2. *Does the autoencoder offer significant processing-efficiency advantages as datasets scale?*
3. *What drawbacks does the autoencoder suffer?*

4.1 Dataset

The experiments were conducted on the 20 Newsgroups [1] dataset - as its known structure serves as aid in diagnosing performance. The dataset consists of 18000 documents that consist of a diverse variety of topics ranging including 'medical', 'space', 'religion', 'computer science' and many more. The corpus was vectorized to a bag-of-words representation after some standard pre-processing including: removing stopwords, lemmatizing, tokenization pruning uncommon or overly common words. Finally, the vocabulary was reduced to only contain 1000 of the most frequent words.

4.2 Model architecture

An LDA model with w tokens, D documents and K topics requires learning two matrices. $\underset{D\times K}{\theta}$ describing the topic distribution over documents and $\underset{K\times V}{\beta}$ portraying the word distribution over topics. To learn the topic distribution over documents with an autoencoder we specify the lowest dimensional hidden layers to correspond with the dimensionality of a K dimensional simplex to soft cluster the D documents over the K topics, that is of dimensions $K-1$. However since we want to learn distributions over the topics - leveraging a standard Gaussian - so we specify dimensionality $2\times(K-1)$ to represent a mean μ and variance σ for each topic. The model is constructed with 100 hidden layers of these dimensions. A uniform Dirichlet prior is specified.

All experiments were conducted on a 2012 Macbook Air with 1,8 GHz Intel Core i5 processor and 4GB of memory. All relevant code is well documented and available here: https://www.zachwolpe.com/research.

4.3 Evaluation metrics

Perplexity is an intrinsic evaluation metric for topic models and an indication of 'how surprised' the model is to see the new document. Recent studies have shown that perplexity scores and human judgement of topics are often not correlated, i.e. high perplexity scores might not yield human interpretable topics [3]. An alternative metric is topic coherence which attempts to quantify how logical (coherent) topics are by measuring the conditional probability of words given a topic. All flavours of topic coherence follow the general equation [11]:

$$\text{Coherence} = \sum_{i<j} score(w_i, w_j)$$

Where words $W = w_1, w_2, ..., w_n$ are ordered from most to least frequently appearing in the topic. The two leading coherence algorithms (UMass and UCI) essentially measure the same thing [13] and we have decided on UMass. The UMass *scores* between $\{w_i, w_j\}$ combinations (which are summed subsequent to calculation) are computed as:

$$score^k_{UMass}(w_i, w_j|K) = \log \frac{D(w_i, w_j) + \epsilon}{D(w_i)}$$

Where K_i is i^{th} topic returned by the model and w_i is more common than w_j $(i < j)$. $D(w_i)$ is the probability of a word w_i is in a document (the number of times w_i appears in a document divided by total documents). $D(w_i, w_j)$ is the conditional probability that w_j will occur in a document, given w_i is in the document - which eludes to some sort of dependency between key words within a topic [11]. ϵ simply provides a smoothing parameter.

4.4 Results

Topic model coherence was used as the primary metric in assessing model performance. To better generalize the findings we compute coherence for a variety of topics K - measuring the performance as models grow in complexity. Further, to account for the sampling distribution coherence was repeatedly computed (10 times) for the same model with difference random samples. It is apparent from Figure 3 that although the autoencoder matches the VI's performance under simple textbook conditions ($K = 5$ topics); as models grow in complexity and dimensionality the autoencoder's coherence scores steadily decline - note that the labels reading *pymc3* and *sklearn* correspond to the AEVB and VI implementations respectively. Although LDA is an unsupervised model, the 20Newsgroups dataset is labelled. So we have the advantage of knowing the true structure of the dataset to be $K = 20$ - which coincides with the best performance using VI. Tables 1 and 2 provides topic examples of both algorithms respectively. The repetive top words in topics 1 and 2 in Table 2 confirms the AEVB's inability to produce meaningful topics.

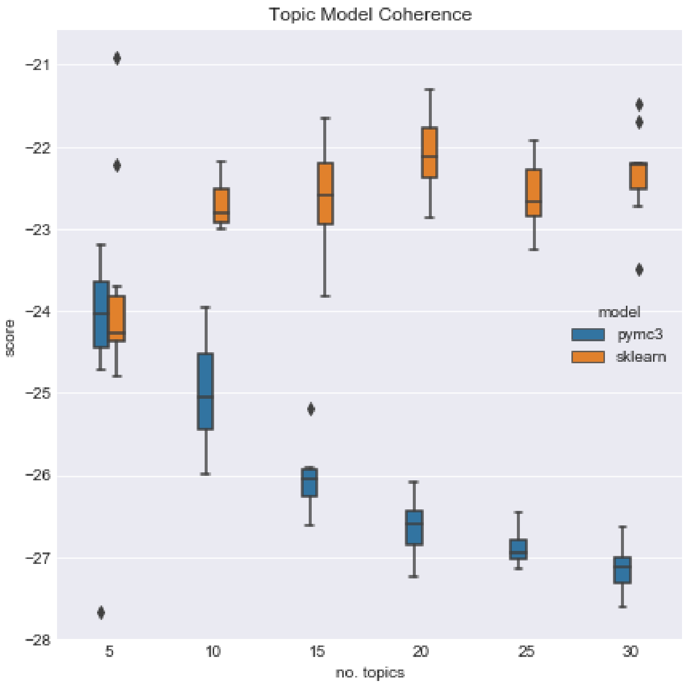

Fig. 3. Topic model coherence achieved for various number of topics, contrasting VI LDA and AEVB LDA implementation.

Table 1. Examples of some topics from LDA VI

Topic 1	Topic 2	Topic 3	Topic 4
mr	space	key	god
law	nasa	car	people
people	data	chip	does
government	program	keys	jesus

Table 2. Examples of some topics from AEVB LDA

Topic 1	Topic 2	Topic 3	Topic 4
know	edu	know	people
just	space	don	god
like	com	like	think
don	information	just	don

5 Discussion on the autoencoder's performance

A callback function was written to assess the autoencoder's iterative process. The callback samples the $\underset{D\times K}{\theta}$ distribution per epoch for the purpose of understanding how the autoencoder allocates documents to topics. Since θ is learnt as a posterior distribution, we need to sample from θ to assess its current state. This was performed under two variations:

1. 1 sample per epoch: to assess randomness.
2. $10'000$ samples per epoch: to assess the current state of θ.

This was conducted for the first 100 epochs to assess the initial conditions of the autoencoder (with a constant $K = 10$ latent topics). Figure 4 depicts the topic allocation for a random sample of documents - coloured by the probability density. One would expect the distribution over topics to begin uniformly spread and thereafter narrow to a single topic. However we observe the vast majority of documents are even initially biasedly assigned to the first few topics, with an increasing density over the first 100 epochs cementing the allocation, despite a uniform Dirichlet prior.

Inspection of the single θ sample depicted in Figure 5 would ideally be completely randomized under initial conditions, however it is apparent it yields some bias. Numerically we can assess the distribution of topic allocations in Figure 4. One would expect a far more uniformly spread distribution over the K topics, with the autoencoder exhibiting bias as early as 5 epochs into running the model; which is then exaggerated.

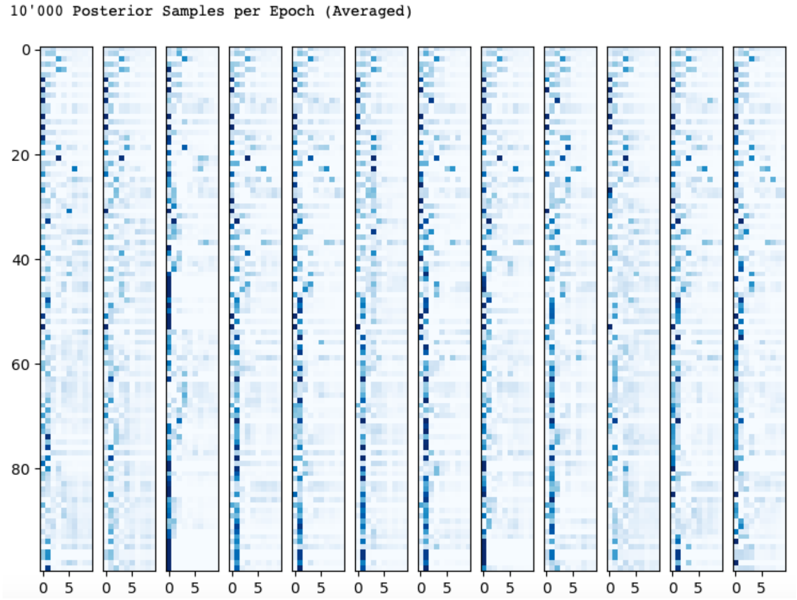

Fig. 4. 12 documents (of the 128 used for the first training batch) are randomly selected. Each column represents a documents - ranging vertically from $Epoch_1$ to $Epoch_{100}$. Each column ranges horizontally from $Topic_0$ to $Topic_{10}$ - since 10 topics are learnt in this model specification.

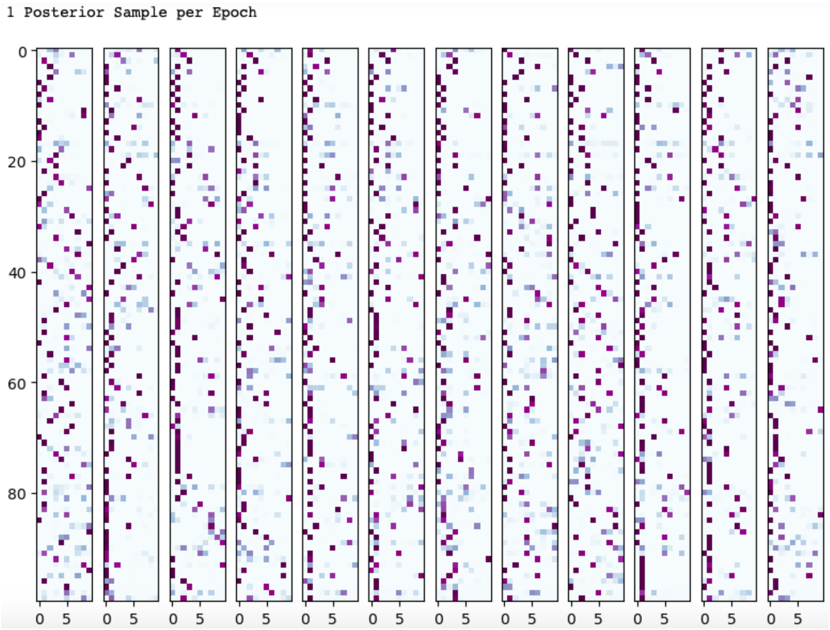

Fig. 5. A single θ sample per epoch.

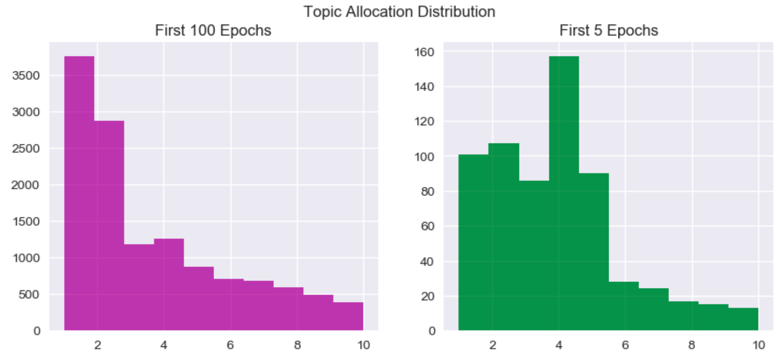

Fig. 6. Distribution of topic allocation over 100 and 5 epochs respectively.

6 Conclusion

AEVB for LDA boasts the theoretical advantage of brisk computation, however this allure evades comparisons of quality. In conducting this analysis our contribution is two fold:

- Perform a thorough comparison between AEVB and VI posterior inference for LDA, uncovering the benefits and drawback of the AEVB as an alternative, speedy, approach.
- Uncover AEVB's shortcomings in an attempt to deduce bias that limits its performance - in addressing *why* and *where* it fails.

After detailed experimentation it is readily apparent that the autoencoder falls short when juxtaposed against established, well engineered techniques. When analyzing the topic coherence performance the autoencoder offers analogous results to the VI LDA for simple models, however, as models grow in complexity it is unambiguously inferior to established methods - when accounting for sampling variability. Results are only comparable under textbook conditions. We show that the encoder fails to adequately explore the domain spaces and heavily biases initial random conditions. Lower predictive accuracy was also found in the PyMC3 tutorial mentioned earlier (https://docs.pymc.io/notebooks/lda-advi-aevb.html). More specifically, in this study the log-likelihood function for topics on held-out words was use as a goodness-of-fit test.

Future work includes an implementation of the solutions offered in [15] to the AEVB's poor performance, namely, the collapsing of the latent variable z and an Laplace approximation to the Dirichlet prior. Furthermore, we want to capitalize on the main proposition of ADVI approaches [8]: that the derivation of closed forms updates is not needed given the variational posterior. More specifically, we plan to apply AEVB on short text topic models [10].

References

1. 20 newsgroups dataset, http://people.csail.mit.edu/jrennie/20Newsgroups/
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
3. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Advances in neural information processing systems. pp. 288–296 (2009)
4. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The helmholtz machine. Neural computation **7**(5), 889–904 (1995)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
6. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
7. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn. Res. **18**(1), 430–474 (Jan 2017), http://dl.acm.org/citation.cfm?id=3122009.3122023
8. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. The Journal of Machine Learning Research **18**(1), 430–474 (2017)
9. Kullback, S.: Information Theory and Statistics. Wiley, New York (1959)
10. Mazarura, J., De Waal, A.: A comparison of the performance of latent dirichlet allocation and the dirichlet multinomial mixture model on short text. In: 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). pp. 1–6. IEEE (2016)
11. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the conference on empirical methods in natural language processing. pp. 262–272. Association for Computational Linguistics (2011)
12. Murphy, K.P.: Machine learning : a probabilistic perspective. MIT Press, Cambridge, Mass. (2013)
13. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 100–108. Association for Computational Linguistics (2010)
14. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
15. Srivastava, A., Sutton, C.: Autoencoding Variational Inference For Topic Models. arXiv:1703.01488 [stat] (Mar 2017), http://arxiv.org/abs/1703.01488