Learning VAE-LDA Models with Rounded Reparameterization Trick

Runzhi Tian  
School of EECS  
University of Ottawa, Canada  
rtian081@uottawa.ca

Yongyi Mao  
School of EECS  
University of Ottawa, Canada  
ymao@uottawa.ca

Richong Zhang  
BDBC and SKLSDE  
Beihang University, China  
zhangrc@act.buaa.edu.cn

Abstract

The introduction of VAE provides an efficient framework for the learning of generative models, including generative topic models. However, when the topic model is a Latent Dirichlet Allocation (LDA) model, a central technique of VAE, the reparameterization trick, fails to be applicable. This is because no reparameterization form of Dirichlet distributions is known to date that allows the use of the reparameterization trick. In this work, we propose a new method, which we call Rounded Reparameterization Trick (RRT), to reparameterize Dirichlet distributions for the learning of VAE-LDA models. This method, when applied to a VAE-LDA model, is shown experimentally to outperform the existing neural topic models on several benchmark datasets and on a synthetic dataset.

1 Introduction

Probabilistic generative models are widely used in topic modelling and have achieved great success in many applications (Deerwester et al., 1990)(Hofmann, 1999)(Blei et al., 2003)(Blei and Lafferty, 2006). A landmark of topic models is Latent Dirichlet Allocation (LDA) (Blei et al., 2003), where a document is treated as a bag of words and each word is modelled via a generative process. More specifically, in this generative process, a topic distribution is first drawn from a Dirichlet prior, then a topic is sampled from the topic distribution and a word is drawn subsequently from the word distribution corresponding to the drawn topic. Since its introduction, LDA has shown great power in a large varieties of natural language applications (Wei and Croft, 2006)(AlSumait et al., 2008)(Mehrotra et al., 2013). However, the classical methods of learning LDA, such as variational techniques and collapsed Gibbs sampling, entails high computation complexity in posterior inference (Blei et al., 2003) (Griffiths and Steyvers, 2004), which limits the ability of LDA on modelling large corpus.

Variational AutoEncoder (VAE) or AutoEncoding Variational Bayes (AEVB) (Kingma and Welling, 2013) provides another choice of learning a generative model. Under the VAE framework, a generative model is specified by first drawing a latent vector \( z \) from a prior distribution and then transforming this vector through a neural network, called decoder, which subsequently generates the observation \( x \). Using a variational inference approach, VAE couples the decoder network with another network, called encoder, responsible for computing the posterior distribution of the latent variable \( z \) for each observation \( x \). A key technique of VAE is its “reparameterization trick”, in which sampling from the posterior is performed by sampling a noise variable \( \epsilon \) from some distribution \( p(\epsilon) \) and then transforming \( \epsilon \) to \( z \) using a differentiable function. This technique allows the model to be trained efficiently using back propagation.

The VAE framework significantly alleviates the computational burden of learning a generative model. Therefore, researchers interested in topic modelling are naturally motivated to consider VAE as an alternative approach to learn LDA, exploiting the power and efficiency of deep learning neural networks. This is also the interest of this paper. However, the key limitation in the application of VAE to Dirichlet-based topic models is that the original reparameterization trick in VAE is not applicable to Dirichlet distributions. In this sense, VAE cannot be directly used for learning any Dirichlet-based topic models. To cope with this, the NVDM model (Miao et al., 2016) discards the Dirichlet assumption and build neural topic models based on Gaussian prior. Although such a Gaussian-based topic model achieves a reasonably good performance on perplexity, the topic words they extracted appear to lack human-
interpretability. Additionally the use of Gaussian prior significantly deviates from the desired Dirichlet distribution and arguably has significant room for improvement.

The adoption of the Dirichlet prior plays a central role in topic modelling, since it nicely captures the intuition that a topic is sampled from a sparse topic distribution. Due to the importance of the Dirichlet assumption in topic modelling, ProdLDA (Srivastava and Sutton, 2017) attempts to apply VAE to LDA by constructing a Laplace approximation to the Dirichlet prior in the softmax basis. However, the Laplace approximation is only used to estimate the prior parameters and ProdLDA has essentially a Gaussian VAE architecture where the KL divergence is on Gaussian distributions. The work of (Joo et al., 2019) argues that the Laplace approximation in ProdLDA fails to capture the multimodality nature of Dirichlet distributions. They then propose DirVAE, in which an approximation of the inverse Gamma CDF (Knowles, 2015) is used to reparameterize Gamma distributions. The Dirichlet samples are then constructed by normalizing Gamma random variables. However, the approximation of inverse Gamma CDF is accurate only when the shape parameter of the Gamma distribution is much less than 1 (Knowles, 2015). This in turn limits the application scope of DirVAE.

In this work, we develop a technique, which we call the Rounded Reparameterization Trick (RRT), to reparameterize Dirichlet distributions. The use of RRT enables VAE as an efficient method for learning LDA, based on which we propose a new neural topic model, referred to as “RRT-VAE”.

Experiments on several datasets show that RRT-VAE outperforms NVDM, ProdLDA, and DirVAE. The experimental results strongly demonstrate the applicability of RRT in topic modelling that utilizes VAE.

2 Preliminary

2.1 LDA

In this paper, we refer to LDA broadly as a generative model characterized by first drawing a distribution \( \theta \) over \( k \) topics from a Dirichlet prior \( \text{Dir}(\theta|\hat{\alpha}) \) and then through a function \( f_{\text{dec}} \), or a decoder, transforming \( \theta \) to a distribution \( P \) over a vocabulary of \( n \) words. That is,

\[
\begin{align*}
\theta & \sim \text{Dir}(\theta|\hat{\alpha}) \quad (1) \\
P & := f_{\text{dec}}(\theta; \beta) \quad (2)
\end{align*}
\]

where \( \beta \) is the parameter of the decoder and will be treated as a \( k \times n \) matrix throughout this paper, although other options are also possible. Under this model, the words in a document is regarded as being drawn i.i.d from this distribution \( P \).

In the classical LDA model (Blei et al., 2003), each row of \( \beta \) represents a word distribution, and the decoder can be written as

\[
f_{\text{dec}}(\theta; \beta) = \theta^T \beta
\]

In the deep learning paradigm, the decoder may be constructed differently, for example,

\[
f_{\text{dec}}(\theta) = \theta^T \text{Softmax} (\beta) \quad (4)
\]

and

\[
f_{\text{dec}}(\theta) = \text{Softmax} (\theta^T \beta) \quad (5)
\]

where in both cases, the rows of \( \beta \) are unconstrained. Note that (4), presented in (Srivastava and Sutton, 2017) is merely a different parameterization of (3) and will be referred to as the “standard decoder” in this paper. The structure in (5), referred to as “product of experts” in (Srivastava and Sutton, 2017), will be called “prod decoder” for simplicity.

2.2 VAE-LDA

The difficulty in learning an LDA model lies in the exact inference of \( \theta \). In the classical LDA, exact inference is replaced by approximation methods using a symbolist variational method (Blei et al., 2003) or MCMC (Griffiths and Steyvers, 2004). In the deep learning era, the development of Variational AutoEncoder (Kingma and Welling, 2013), a connectionist counterpart of the symbolist variational methods, provides an alternative approach to handle this difficulty.

When applying VAE to an LDA model, the model is augmented with an encoder network \( f_{\text{enc}} \). Specifically, the encoder takes as the input the bag-of-words (i.e., word histogram) representation \( x \) of a document and outputs a \( k \)-dimensional parameter \( \alpha \), and then the Dirichlet distribution with parameter \( \alpha \) is taken as the posterior distribution \( q(\cdot|\alpha) \) of \( \theta \):

\[
\begin{align*}
\alpha & := f_{\text{enc}}(x; \Pi) \quad (6) \\
q(\cdot|\alpha) & := \text{Dir}(\cdot|\alpha) \quad (7)
\end{align*}
\]
where \( \Pi \) denotes the parameters of the encoder.

Under the VAE framework, the parameters of the encoder and the decoder are jointly optimized by minimizing the negative Evidence Lower Bound (ELBO):

\[
\mathcal{L}(\Pi, \beta; x) = \text{KL}(q(\theta|\alpha)||p(\theta|\hat{\alpha})) - \mathbb{E}_{q(\theta|\alpha)}[J(\theta, x)]
\]

(8)

where \( p(\theta|\hat{\alpha}) := \text{Dir}(\theta|\hat{\alpha}) \), the Dirichlet prior; and

\[
J(\theta, x) := x^T \log f_{\text{dec}}(\theta)
\]

(9)

We refer to the model specified by the loss function (8) as VAE-LDA.

Note that the KL term in (8) has a closed-form expression

\[
\text{KL}(q(\theta|\alpha)||p(\theta|\hat{\alpha})) = \log \Gamma \left( \sum \alpha_i \right)
\]

\[
- \sum \log \Gamma(\alpha_i) - \log \Gamma \left( \sum \hat{\alpha}_i \right) + \sum \log \Gamma(\hat{\alpha}_i)
\]

\[
+ \sum (\alpha_i - \hat{\alpha}_i) \left( \psi(\alpha_i) - \psi \left( \sum \alpha_i \right) \right)
\]

The gradient of this term can be obtained directly. The optimization of the second term in (8) is however challenging, since it has no closed-form expression. Additionally, when using a stochastic approximation, one must deal with back-propagating gradient signals through a sampling process.

One way to deal with this is to use a score function estimator (Williams, 1992)(Glynn, 1990). But such an approach is known to give rise to high variances in the gradient estimation, due to which a reliable estimate would require drawing a large number of \( \theta \) from the posterior \( q(\cdot|\alpha) \) and make learning inefficient. In the framework of VAE, a “reparameterization trick” is introduced as an elegant solution to such a problem, where the posterior is reparameterized as drawing a noise from another distribution and re-expressing the posterior as a differentiable function of the noise. However when the posterior distribution is a Dirichlet distribution (or a related distribution such as Beta and Gamma distributions), no such noise distribution and continuous functions are known to exist. Thus the standard reparameterization trick does not apply to learning VAE-LDA.

3 Rounded Reparameterization Trick

To tackle the limitation of the standard reparameterization trick, we propose a new reparameterization method, referred to as rounded reparameterization trick or RRT.

Given a real number \( \Delta \), we define a “\( \Delta \)-rounding” function \( \lfloor \cdot \rfloor_\Delta \) as follows: For any real number \( a \),

\[
[a]_\Delta = \left\lfloor \frac{a}{\Delta} \right\rfloor \cdot \Delta
\]

(10)

where the operation \( \lfloor \cdot \rfloor \) is the integer floor (or “rounding down”) operation. For example, \( \lfloor 3.14159265 \rfloor_{0.001} = 3.141 \). When the \( \Delta \)-rounding operation applies to a vector, it acts on the vector component-wise.

In RRT, we draw an auxiliary variable \( \hat{\theta} \) from a “rounded” posterior distribution \( q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta) \).

\[
\hat{\theta} \sim q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta)
\]

(11)

and compute

\[
\hat{\theta} = g(\hat{\theta}; \alpha) := \hat{\theta} + \lambda (\alpha - \lfloor \alpha \rfloor_\Delta)
\]

(12)

Then \( \hat{\theta} \) is used to approximate \( \theta \sim q(\theta|\alpha) \). In (12), the parameter \( \lambda \) is a hyper parameter which will serve to adjust the strength of the gradient. Note that when choosing a very small rounding precision \( \Delta \), we expect that the distribution \( q(\cdot|\alpha) \) of \( \theta \) and the distribution \( q(\cdot|\alpha) \) are nearly identical. As a consequence, \( \mathbb{E}_{q(\theta|\alpha)}[J(\theta, x)] \) and its replacement \( \mathbb{E}_{q(\hat{\theta}|\alpha)}[J(\theta, x)] \) are also very close to each other. Thus such a replacement keeps the loss function very close to the original loss in (8).

For shorter notations, we denote

\[
A(\alpha) := \mathbb{E}_{q(\hat{\theta}|\alpha)}[J(\theta, x)]
\]

(13)

\[
\tilde{A}(\alpha) := \mathbb{E}_{q(\hat{\theta}|\alpha)}[J(\theta, x)]
\]

(14)

and

\[
\tilde{\mathcal{L}}(\Pi, \beta; x) := \text{KL}(q(\theta|\alpha)||p(\theta|\hat{\alpha})) - \tilde{A}(\alpha)
\]

(15)

Constructing gradient estimator using RRT

The gradient \( \nabla_\alpha \tilde{A}(\alpha) \) can be expressed as a sum of two terms:

\[
\nabla_\alpha \tilde{A}(\alpha) = \nabla_\alpha \mathbb{E}_{q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta)} \left[ J \left( g(\hat{\theta}, \alpha), x \right) \right]
\]

\[
= \nabla_\alpha \int q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta) J \left( g(\hat{\theta}, \alpha), x \right) d\hat{\theta}
\]

\[
= \int \nabla_\alpha q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta) J \left( g(\hat{\theta}, \alpha), x \right) d\hat{\theta}
\]

\[
+ \int q(\hat{\theta}|\lfloor \alpha \rfloor_\Delta) \nabla_\alpha J \left( g(\hat{\theta}, \alpha), x \right) d\hat{\theta}
\]

The first term in sum is usually estimated through the score function estimator. But this is unnecessary in this case. To see this, note that \( \nabla_\alpha \lfloor \alpha \rfloor_\Delta = \)
0 almost everywhere. This implies that the first term is in fact 0 at every \( \alpha \) for which the gradient exists. The next lemma then immediately follows.

**Lemma 1** For any \( \alpha \) at which the gradient \( \nabla_\alpha \tilde{A}(\alpha) \) exists,

\[
\nabla_\alpha \tilde{A}(\alpha) = \lambda \mathbb{E}_{q(\theta|\alpha, \Delta)} \left[ \nabla_\theta J(\theta, x)|_{\theta = g(\theta, \alpha)} \right]
\]

The fact that the score function estimator is not very few samples in Monte-Carlo estimation.

Using Lemma 1, one can directly express the stochastic (Monte Carlo) estimate of the gradient \( \nabla_\alpha \tilde{A}(\alpha) \) as

\[
\nabla_\alpha \tilde{A}(\alpha) \approx \frac{\lambda}{N} \sum_{i=1}^{N} \nabla_\theta J(\theta, x)|_{\theta = g(\theta_i, \alpha)}
\]

where \( \theta \sim q(\theta|\alpha, \Delta) \). The fact that \( q \) is differentiable almost everywhere with respect to \( \alpha \) allows the gradient signal to back propagate and can be implemented using automatic differentiation libraries.

Due to the low variance in this estimator, it is sufficient to sample only a single \( \theta \) from \( q(\theta|\alpha, \Delta) \), namely, take \( N = 1 \) in (16).

At this end, we conclude that the loss function \( \tilde{\mathcal{L}} \) obtained by replacing \( \theta \) with \( \tilde{\theta} \) is very close to the original loss function \( \mathcal{L} \), and a low-variance gradient estimator can be easily constructed from \( \tilde{\mathcal{L}} \). This completes the description of RRT.

**On the discontinuities induced by RRT**

Notably the \( \Delta \)-rounding function in RRT induces discontinuities in the resulting loss function \( \tilde{\mathcal{L}} \). This is because \( \tilde{A}(\alpha) \) is discontinuous in \( \alpha \) and countably many discontinuity points exist. One may be concerned with whether an update of \( \alpha \) may “hop over” a discontinuity point of \( \tilde{A}(\alpha) \) and cause training unstable or diverge.

To that end, we have the following result.

**Lemma 2** Suppose that \( J \) is \( \zeta \)-lipschitz in \( \theta \) and \( A(\alpha) \) is \( \gamma \)-lipschitz in \( \alpha \). Then for any integer \( m \),

\[
|\tilde{A}(m\Delta) - \tilde{A}(m\Delta - \epsilon)| < (\gamma + \zeta \lambda)\Delta
\]

when \( \epsilon \to \Delta \).

We note that when \( \epsilon \to \Delta \), the quantity \( |\tilde{A}(m\Delta) - \tilde{A}(m\Delta - \epsilon)| \) measures the magnitude of a sudden rise or drop when an update hops over the discontinuity point \( \alpha = m\Delta \). When this magnitude is small, the discontinuity causes little impact on the stability of training. The upper bound of this quantity given by this lemma suggests that as long as \( J(\theta) \) and the objective function \( A(\alpha) \) are reasonably smooth, one may control this magnitude to be small by choosing a relatively small \( \Delta \). On the other hand, in case one indeed chooses a relatively large \( \Delta \), the bound of this magnitude may become quite large. However in this case, the update will have much smaller chance of hopping over a discontinuity point, and one still expects no serious problem caused by these discontinuities.

We now present the proof.

**Proof:** Clearly, \( \tilde{A}(m\Delta) = A(m\Delta) \). And

\[
\tilde{A}(m\Delta - \epsilon) = \mathbb{E}_{q(\theta|(m-1)\Delta)} J(\theta + \lambda(\Delta - \epsilon))
\]

\[
= \mathbb{E}_{q(\theta|(m-1)\Delta)} \{ J(\theta) + \lambda(\Delta - \epsilon)J'(\theta) \}
\]

\[
= A((m - 1)\Delta) + \lambda(\Delta - \epsilon) \cdot \mathbb{E}_{q(\theta|(m-1)\Delta)} J'(\theta)
\]

Since \( J \) is \( \zeta \)-lipschitz,

\[
A((m - 1)\Delta) - \zeta \lambda(\Delta - \epsilon)
\]

\[
< \tilde{A}(m\Delta - \epsilon) < A((m - 1)\Delta) + \zeta \lambda(\Delta - \epsilon)
\]

It follows

\[
\tilde{A}(m\Delta) - A((m - 1)\Delta) + \zeta \lambda(\Delta - \epsilon)
\]

\[
> \tilde{A}(m\Delta) - \tilde{A}(m\Delta - \epsilon)
\]

\[
> A(m\Delta) - A((m - 1)\Delta) - \zeta \lambda(\Delta - \epsilon)
\]

Since \( A(\cdot) \) is \( \gamma \)-lipschitz, then

\[
\gamma \Delta + \zeta \lambda(\Delta - \epsilon)
\]

\[
> \tilde{A}(m\Delta) - \tilde{A}(m\Delta - \epsilon) > -\gamma \Delta - \zeta \lambda(\Delta - \epsilon)
\]

It follows

\[
|\tilde{A}(m\Delta) - \tilde{A}(m\Delta - \epsilon)| < (\gamma + \zeta \lambda)\Delta
\]

This proves the lemma. \( \square \)

It is clear that when \( \Delta \) is small, the discontinuity is not obvious and has small impact on the optimization of the model.

**4 Related Work**

Beyond topic modelling, another theme of research related to this work is the estimation of gradient in neural networks containing stochastic nodes or samplers. In this setting, one desires that the gradient signal is capable of back-propagating through
the samplers. A classical method for this purpose is to construct a score function estimator, also known as the “log derivative trick” or REINFORCE (Williams, 1992) (Glynn, 1990). However, despite giving an unbiased estimate, the Monte-Carlo implementation of such an estimator typically suffers from a high variance, and thus relies on some additional variance-reduction techniques (Greensmith et al., 2004). Reparameterization trick (Kingma and Welling, 2013), as mentioned above, may also be used to back-propagate gradients through samples and enjoys a low-variance advantage. Unfortunately this technique is not applicable to many distributions such as Gamma, Beta and Dirichlet distributions. Various efforts have been spent on extending the applicability of reparameterization trick to a broader range. These works include, for example, G-REP (Ruiz et al., 2016), RSVI (Naesseth et al., 2016) and Implicit Reparameterization Gradients (Figurnov et al., 2018), etc. These methods usually involve complicated gradient derivations and are often difficult to implement in neural networks.

5 Experiments and Results

To quantitatively evaluate RRT-VAE, we conduct experiments on synthetic datasets and five real-world datasets. Our model is compared with several existing topic models: Online LDA (Hoffman et al., 2010), NVDM (Miao et al., 2016), ProdLDA (Srivastava and Sutton, 2017) and DirVAE (Joo et al., 2019).

In the experiments, we adopt three MLPs with ReLU activations as the encoder of RRT-VAE, where each hidden layer is set to 500 dimensions. We apply an exponential function on the outputs of the encoder, so that the outputs are positive values. The topic distribution vectors are sampled through RRT and then normalized before being passed to the decoder. For Online LDA, we use the standard implementation from scikit-learn (Pedregosa et al., 2011). The encoder structures of NVDM, ProdLDA and DirVAE are built according to (Miao et al., 2016), (Srivastava and Sutton, 2017) and (Joo et al., 2019), where in our experiments the dimension of each hidden layer is set to 500.

On the real-world datasets, we adopt the prod decoder, which is examined to be superior to the prod decoder on this learning task (see Appendix A.1).

5.1 Datasets

Synthetic datasets. We construct three synthetic datasets based on the LDA generative process: a \(30 \times 500\) topic-word probability matrix \(\beta_g\) is generated as the ground truth; each dataset is then generated based on \(\beta_g\) using different Dirichlet priors \(\alpha_g \cdot 1 \in \mathbb{R}^{30}\), where \(1\) denotes the all-one vector. We set \(\alpha_g\) to \([0.01, 0.05, 0.1]\) for the three datasets and the vocabulary size to 500. Each dataset has 20000 training examples.

Real-world datasets. We use five real-world datasets in our experiments: 20NG, RCV1-v2, AGNews, DBPedia (Lehmann et al., 2015), and Yelp review polarity (Zhang et al., 2015). The 20NG and RCV1-v2 datasets are the same as (Miao et al., 2016). The other three datasets are preprocessed through tokenizing, stemming, lemmatizing and the removal of stop words. We keep the most frequent 2000 words in DBPedia and Yelp. For AGNews, we keep the words which are contained in no more than half the documents and are contained in at least 15 documents. The statistics of the cleaned datasets are summarized in Table 1.

|          | 20NG | AGNews | RCV1-v2 | DBpedia | Yelp |
|----------|------|--------|---------|---------|------|
| #Train   | 11258| 120000 | 794414  | 560000  | 560000|
| #Test    | 7487 | 7600   | 10000   | 70000   | 38000|
| #Vocab   | 1995 | 10630  | 10000   | 20000   | 20000|

Table 1: Summary of different datasets

5.2 Evaluation Methods

On the real-world datasets, we use perplexity and normalized pointwise mutual information (NPMI) (Lau et al., 2014) as the evaluation metrics. On synthetic datasets, we propose topic words recovery accuracy (or “recovery accuracy” in short) to evaluate the model performance.

Specifically, we extract the top-10 highest-probability word indexes from each row of \(\beta_g\). The standard decoder.

\(^2\)As reported in (Srivastava and Sutton, 2017), ProdLDA also appears to extract many repetitive words when using the standard decoder.

\(^3\)For 20NG and RCV1-v2, we use the datasets provided by https://github.com/ysmiao/nvdm

\(^4\)http://groups.di.unipi.it/gulli/AG_corpus_of_news_articles.html
extracted word indexes constitute a $30 \times 10$ topic-word matrix $T_g$. Our goal is to use the topic models to recover this matrix. Denote by $T_L$, a matrix extracted from the learned $\beta$ matrix of a model. Note that the rows of $T_L$ are arbitrarily ordered. To count how many words in the $i$th row $t_g^{(i)}$ of $T_g$ is recovered in a topic in $T_L$, we compare $t_g^{(i)}$ with each row in $T_L$. We count the number of common words in the compared two rows and keep the maximum count as the number of recovered words in $t_g^{(i)}$. The recovery accuracy is then defined as the total number of recovered words in all rows of $T_L$ divided by the total number of words.

We note that after a row of $T_g$ is compared with $T_L$ as the target of coverage, the found best-matching row in $T_L$ is not removed. This approach is better than the alternative approach of greedily removing the best-matching row, since the latter would give an accuracy result that depends on the row ordering in $T_g$. Additionally we note that the data generation process assures that the rows of $T_g$ each contain 10 distinct words. For this reason, keeping the found best-matching row in $T_L$ in each step entails no problem.

5.3 Influence of Parameter Settings

In this section, we run RRT-VAE on 20NG and the synthetic datasets to explore its performance under different parameter settings.

5.3.1 Results on 20NG

Prior settings. Prior settings are claimed to have a significant influence on model performance (Wallach et al., 2009). In this experiment, we run RRT-VAE on the 20NG dataset using four symmetric Dirichlet prior settings $[0.02, 0.2, 1.0, 2.0]$. The number of topics is set to 50 and $\lambda$ is set to 0.01 in all experiments. We use $\Delta = 10^{-10}$ as the rounding precision such that accurate Dirichlet samples can be drawn.

![Figure 1: Training performance of RRT-VAE with different prior (left) and $\lambda$ settings (right).](image)

As shown in Figure 1 (left), when using a larger prior parameter (1 or larger), the training loss drops more rapidly and converges to a lower value. Table 2 reports the corresponding testing results. We found that when using a smaller prior setting, RRT-VAE tends to achieve a better topic coherence (NPMI) while sacrificing some performance on perplexity. One possible explanation of these phenomena is that a smaller prior setting (lower than 1) encourages the encoder network to sample a sparser topic distribution $\theta$. The sparsity of $\theta$ in turn makes it easier for the model to assign a very small probability on some existing words in a document and thus increases the training loss and perplexity.

To verify this conjecture, we construct a simple method to measure sparsity: after the training, we randomly feed 1000 training samples into the encoder network and obtain 1000 topic distribution vectors $\{\theta_i\}_{i=1}^{1000}$. For each $\theta_i$, we calculate the difference between its largest and smallest probability value and then average these differences over the 1000 samples. Clearly, a larger difference value indicates a sparser $\theta$, e.g. the maximum difference 1 is achieved by a one-hot vector. From the sparsity measurements in Table 2, we see that a smaller prior setting causes the encoder to generate sparser topic distribution vectors, which in turn hinders the convergence of the training loss to a lower value and hence causes a higher perplexity. On the other hand, sparser topic distributions tend to improve NPMI, although this improvement is slight.

$\lambda$ settings. The “gradient control” parameter $\lambda$ in RRT adjusts the strength of the gradient signal back-propagated to the encoder while also influencing the variance of the Monte Carlo gradient estimator. Figure 1 (right) and Table 3 report the influence of different $\lambda$ settings on the model performance, where the number of topics is set to 50.
and the prior is set to 1. As shown, when \( \lambda \) is set too small (e.g. \( \lambda = 0.001 \)), the training loss fails to converge to a lower value, resulting in a higher perplexity and worse NPMI. The best performance is achieved when \( \lambda \) is set between around 0.01 and 0.005. Different \( \lambda \) settings can bring similar training performances but different testing results. For example, when \( \lambda \) is set to 0.1 and 0.01, the corresponding training performances are very similar (see Figure 1 (right), blue and grey dash line), however, \( \lambda = 0.01 \) achieves a better perplexity and NPMI result.

5 The corresponding perplexity value (green line in Figure 2 (b)) also increases from 1078 to around 1400. In contrast, the sparsity levels of \( \hat{\alpha} = 1.0 \) and \( \hat{\alpha} = 2.0 \) remain unchanged. Their corresponding perplexities also stay at the same levels.

5.3.2 Results on Synthetic datasets

Our experiments on the synthetic datasets again demonstrate that the rounding precision has little impact on the training stability. Figure 3 (left) exhibits how different \( \Delta \) settings influence the training performance of RRT-VAE when \( \alpha_g = 0.01 \) (the results of \( \alpha_g = 0.05 \) and 0.1 are shown in Appendix A.2). As shown, all the training losses decrease stably, although a higher \( \Delta \) setting hinders the loss converging to a lower value. Figure 3 (right) reports how different \( \Delta \) settings influence the recovery accuracy of RRT-VAE on three synthetic datasets. It can be seen that a smaller \( \Delta \) achieves a better performance. Specifically, when \( \Delta = 1 \), the training loss remains at a high value and the corresponding recovery accuracy is lower than 60%, indicating that RRT-VAE fails to fit the true data distribution. In contrast, when \( \Delta = 10^{-10} \), RRT-VAE fits the data well: the training loss drops rapidly and converges to a much lower value; the resulting recovery accuracy reaches up to 90%.

Recall that on 20NG, both the training and testing performances are insensitive to the rounding precision. In contrast, on synthetic datasets, the rounding precision has a significant influence. This phenomenon is reasonable, since the synthetic data strictly satisfies the LDA generative process.

Since the sparsity we define is computed from the randomly sampled \( \theta \), it is inherently stochastic due to randomness in \( \theta \). Thus a small fluctuation of the computed sparsity value needs not to indicate a true difference of sparsity levels. For example, on the green line of Figure 2 (d), the sparsity values of \( \Delta = 0.01 \) and \( \Delta = 10^{-10} \) are different, but the difference is not large enough to suggest that the two models have different sparsity levels; such a difference is primarily due to stochastic irregularity in our sparsity computation.
higher $\Delta$ setting causes the rounded distribution deviate the Dirichlet posterior, thereby interfering with the fitting of the data. On the other hand, the underlying distribution of the real-world data does not strictly conform to the LDA assumption. This deviation, therefore, has little impact on fitting the data.

5.4 Comparison with Other Models

In this section, we compare RRT-VAE with other existing topic models on both real-world datasets and synthetic datasets.

Real-world datasets

On real-world datasets, we do not compare Online LDA, since the training of Online LDA on large datasets is extremely time consuming and Online LDA fails to obtain any good results after being trained for a long time (results of Online LDA on 20NG are shown in Appendix B.2). For ProdLDA, DirVAE and RRT-VAE, we tune the prior parameter from $[0.02, 0.2, 1.0]$. The best $\lambda$ settings of RRT-VAE for each dataset are shown in Table 4. All the compared models adopt the same prod decoder of (5) on the real-world datasets.

| Dataset | NVDM | ProdLDA | DirVAE | RRT-VAE |
|---------|------|---------|--------|---------|
| 20NG    | 773/0.152 | 987/0.262 | 970/0.277 | 978/0.271 |
| AGNews  | 1067/0.086 | 1457/0.196 | 1573/0.287 | 1318/0.287 |
| RCV1-v2 | 511/0.121 | 623/0.160 | 746/0.137 | 623/0.262 |
| DBPedia | 617/0.093 | 1065/0.101 | 1018/0.102 | 851/0.227 |
| Yelp    | 1003/0.120 | 1244/0.064 | 1353/0.068 | 1251/0.266 |

Table 4: Optimal $\lambda$ settings of RRT-VAE for different datasets.

Table 5: Perplexity/NPMI of the compared topic models on five datasets. The number of topic is set to 50.

| Dataset | NVDM | ProdLDA | DirVAE | RRT-VAE |
|---------|------|---------|--------|---------|
| 20NG    | 1167/0.140 | 1050/0.172 | 973/0.215 | 997/0.214 |
| AGNews  | 1160/0.056 | 2434/0.024 | 1523/0.156 | 1914/0.226 |
| RCV1-v2 | 482/0.107 | 604/0.085 | 706/0.045 | 669/0.254 |
| DBPedia | 597/0.055 | 997/0.113 | 1028/0.041 | 884/0.161 |
| Yelp    | 996/0.069 | 1272/0.072 | 1259/0.044 | 1325/0.174 |

Table 6: Perplexity/NPMI of the compared topic models on five datasets. The number of topic is set to 200.

In general, perplexity measures the goodness-of-fit of data to a learned model under the maximum likelihood principle. This makes it a valid metric for evaluation when the learning objective (as in the considered models) aims at maximizing the data likelihood. On the other hand, we note that traditionally in all VAE-LDA models (e.g., those compared in this paper) and also in this paper, perplexity is in fact approximately computed using the evidence lower bound (ELBO) of the data likelihood, since exact computation of the data likelihood is usually intractable. But the perplexity computed this way aggregates the overall effects of both the learned decoder (i.e., the $\beta$ matrix) and the learned encoder. Therefore it does not provide a direct evaluation of the learned word distributions in the $\beta$ matrix. This problem is overcome by the additional NPMI measure, which is computed directly from the $\beta$ matrix and serves as a more indicative quality measurement of the learned topics.

Table 7: Topic words extracted from the Yelp dataset. From top to bottom, each cell is extracted by NVDM, ProdLDA, DirVAE and RRT-VAE. More examples are exhibited in Appendix B.3.
be easily inferred that the corresponding topics are “trade”, “Japanese food”, “medical” and “fitness”. But it is not the case for the other models.

**Synthetic datasets**

We compare RRT-VAE with Online LDA, ProdLDA and DirVAE on three synthetic datasets which are generated by different Dirichlet parameters. The compared three neural topic models adopt the same standard decoder of (4). Since NVDM is a pure Gaussian VAE model without any approximation of Dirichlet distributions, it is not compared in this experiment. Table 8 reports the recovery accuracy of the compared models. The experimental results strongly demonstrate the ability of RRT-VAE as an inference method to learn LDA. Specifically, RRT-VAE levels with Online LDA on recovery accuracy, while it enjoys a much higher computational efficiency. Among three neural topic models, RRT-VAE clearly outperforms the others. Appendix A.3 shows an example of the ground truth matrix $T_g$ and the matrix recovered by RRT-VAE.

|                  | Online LDA | ProdLDA | DirVAE | RRT-VAE |
|------------------|------------|---------|--------|---------|
| $\alpha_g=0.01$  | 87.33%     | 84.0%   | 91.33% | **96.67%** |
| $\alpha_g=0.05$  | 91.33%     | 83.0%   | 84.67% | **93.0%** |
| $\alpha_g=0.1$   | 90.0%      | 55.67%  | 83.67% | **91.0%** |

Table 8: Recovery accuracy of four topic models on synthetic datasets generated by three different $\alpha_g$ settings. For RRT-VAE, $\lambda$ is set to 1; $\Delta$ is set to $10^{-10}$.

### 6 Concluding Remarks

In this paper, rounded reparameterization trick, or RRT, is shown as an effective and efficient reparameterization method for Dirichlet distributions in the context of learning VAE based LDA models. In fact, the applicability of RRT can be generalized beyond Dirichlet distributions. This is because any distribution can be reparameterized to an “RRT form” as long as a sampling algorithm exists for that distribution. Thus it will be interesting to investigate the performance of RRT in other applications of VAE beyond topic modelling. Successes in these investigations will certainly extend the applicability of VAE to much broader application domains and model families.

---

**References**

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. 2008. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *2008 eighth IEEE international conference on data mining*, pages 3–12. IEEE.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Lda. *Journal of machine Learning research*, 3(Jan):993–1022.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. 2018. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452.

Peter W Glynn. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. 2019. Dirichlet variational autoencoder. *arXiv preprint arXiv:1901.02739*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
David A Knowles. 2015. Stochastic gradient variational bayes for gamma approximating distributions. arXiv preprint arXiv:1509.01631.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 530–539.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web, 6(2):167–195.

Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. 2013. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pages 889–892.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In International conference on machine learning, pages 1727–1736.

Christian A Naesseth, Francisco JR Ruiz, Scott W Linderman, and David M Blei. 2016. Reparameterization gradients through acceptance-rejection sampling algorithms. arXiv preprint arXiv:1610.05683.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, pages 100–108. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct):2825–2830.

Francesco R Ruiz, Michalis Titsias RC AUEB, and David Blei. 2016. The generalized reparameterization gradient. In Advances in neural information processing systems, pages 460–468.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. arXiv preprint arXiv:1703.01488.

Hanna M Wallach, David M Mimno, and Andrew McCallum. 2009. Rethinking lda: Why priors matter. In Advances in neural information processing systems, pages 1973–1981.

Xing Wei and W Bruce Croft. 2006. Lda-based document models for ad-hoc retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 178–185.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.

A Additional Results on Synthetic Datasets

A.1 Topic recovery accuracy using prod decoder

| ProdLDA | DirVAE | RRT-VAE |
|---------|-------|--------|
| $\alpha_g=0.01$ | 50.33% | 59.33% | 61.33% |
| $\alpha_g=0.05$ | 48.33% | 64.67% | 59.67% |
| $\alpha_g=0.1$ | 43.0% | 64.66% | 62.33% |

Table 9: Topic words recovery accuracy of three neural topic models on synthetic datasets generated with three different $\alpha_g$ settings. The models adopt the same prod decoder structure. For RRT-VAE, $\lambda$ is set to 1; $\Delta$ is set to $10^{-10}$.

Table 9 reports the topic recovery accuracy of three neural topic models using the prod decoder. Compared to Table 8, it can be seen that the standard decoder significantly outperforms the prod decoder on the synthetic datasets.

A.2 Training performance

Figure 4 plots the training loss curves of RRT-VAE with different $\Delta$ settings on two synthetic datasets ($\alpha = 0.05$ and $\alpha = 0.1$). The curves perform similarly to Figure 3 (left).

A.3 Recovered topic words

Table 10 exhibits an example of the ground truth topic word matrix $T_g$ used in our experiments and
Table 12: The experimental results of Online LDA on 20NG.

| Perplexity | NPMI |
|------------|------|
| 50 topics  | 1.181|
| 200 topics | 0.162|

Table 13: Topic words extracted by RRT-VAE from four different datasets. From top to bottom, each cell is extracted from 20NG, AGNews, RCV1-v2 and DBpedia.