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 ϵ from some distribution $p(\epsilon)$ and then transforming ϵ 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-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1315–1325,
November 16–20, 2020. ©2020 Association for Computational Linguistics

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".[1] 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 θ over k topics from a Dirichlet prior $\mathrm{Dir}\left(\theta|\hat{\alpha}\right)$ and then through a function f_{dec}, or a *decoder*, transforming θ to a distribution P over a vocabulary of n words. That is,

$$\theta \sim \mathrm{Dir}\left(\theta|\hat{\alpha}\right) \tag{1}$$

$$P := f_{\mathrm{dec}}(\theta; \beta) \tag{2}$$

where β 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 β represents a word distribution, and the decoder can be written as

$$f_{\mathrm{dec}}(\theta; \beta) = \theta^T \beta \tag{3}$$

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

$$f_{\mathrm{dec}}(\theta) = \theta^T \mathrm{Softmax}\left(\beta\right) \tag{4}$$

$$\text{and} \quad f_{\mathrm{dec}}(\theta) = \mathrm{Softmax}\left(\theta^T \beta\right) \tag{5}$$

where in both cases, the rows of β 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 θ. 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_{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 α, and then the Dirichlet distribution with parameter α is taken as the posterior distribution $q(\cdot|\alpha)$ of θ:

$$\alpha \;:=\; f_{\mathrm{enc}}(x; \Pi) \tag{6}$$

$$q(\cdot|\alpha) \;:=\; \mathrm{Dir}(\cdot|\alpha) \tag{7}$$

[1] Code will be available at https://github.com/rzTian/RRT-VAE/tree/main

where Π 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):

$$\mathcal{L}(\Pi, \beta; x) = \mathrm{KL}\left(q(\theta|\alpha)||p(\theta|\hat{\alpha})\right) - \mathbb{E}_{q(\theta|\alpha)}\left[J(\theta, x)\right] \tag{8}$$

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

$$J(\theta, x) := x^{\mathrm{T}} \log f_{\mathrm{dec}}(\theta) \tag{9}$$

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

$$\begin{aligned}&\mathrm{KL}(q(\theta|\alpha)||p(\theta|\hat{\alpha})) = \log\Gamma\left(\sum\alpha_i\right)\\&-\sum\log\Gamma(\alpha_i) - \log\Gamma\left(\sum\hat{\alpha}_i\right) + \sum\log\Gamma(\hat{\alpha}_i)\\&+\sum(\alpha_i - \hat{\alpha}_i)\left(\psi(\alpha_i) - \psi\left(\sum\alpha_i\right)\right)\end{aligned}$$

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 θ 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 Δ, we define a "Δ-rounding" function $\lfloor\cdot\rfloor_\Delta$ as follows: For any real number a,

$$\lfloor a\rfloor_\Delta = \left\lfloor\frac{a}{\Delta}\right\rfloor\cdot\Delta \tag{10}$$

where the operation $\lfloor\cdot\rfloor$ is the integer floor (or "rounding down") operation. For example, $\lfloor\ 3.14159265\rfloor_{\Delta=0.001} = 3.141$. When the Δ-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 $q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)$,

$$\hat{\theta}\sim q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right) \tag{11}$$

and compute

$$\widetilde{\theta} = g(\hat{\theta};\alpha) := \hat{\theta} + \lambda\left(\alpha - \lfloor\alpha\rfloor_\Delta\right) \tag{12}$$

Then $\widetilde{\theta}$ is used to approximate $\theta\sim q(\theta|\alpha)$. In (12), the parameter λ is a hyper parameter which will serve to adjust the strength of the gradient. Note that when choosing a very small rounding precision Δ, we expect that the distribution $\widetilde{q}(\cdot|\alpha)$ of $\widetilde{\theta}$ and the distribution $q(\cdot|\alpha)$ are nearly identical. As a consequence, $\mathbb{E}_{q(\theta|\alpha)}[J(\theta,x)]$ and its replacement $\mathbb{E}_{\widetilde{q}(\theta|\alpha)}[J(\theta,x)]$ 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

$$\begin{aligned}A(\alpha) &:= \mathbb{E}_{q(\theta|\alpha)}[J(\theta,x)] &(13)\\ \widetilde{A}(\alpha) &:= \mathbb{E}_{\widetilde{q}(\theta|\alpha)}[J(\theta,x)] &(14)\end{aligned}$$

and

$$\widetilde{\mathcal{L}}(\Pi,\beta;x) := \mathrm{KL}\left(q(\theta|\alpha)||p(\theta|\hat{\alpha})\right) - \widetilde{A}(\alpha) \tag{15}$$

Constructing gradient estimator using RRT

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

$$\begin{aligned}\nabla_\alpha\widetilde{A}(\alpha) =&\nabla_\alpha\mathbb{E}_{q(\hat{\theta}|\lfloor\alpha\rfloor_\Delta)}\left[J\left(g(\hat{\theta},\alpha),x\right)\right]\\ =&\nabla_\alpha\int q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)J\left(g(\hat{\theta};\alpha),x\right)d\hat{\theta}\\ =&\int\nabla_\alpha q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)J\left(g(\hat{\theta};\alpha),x\right)d\hat{\theta}\\ &+\int q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)\nabla_\alpha J\left(g(\hat{\theta};\alpha),x\right)d\hat{\theta}\end{aligned}$$

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 α for which the gradient exists. The next lemma then immediately follows.

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

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

The fact that the score function estimator is not needed for estimating the gradient $\nabla_\alpha \widetilde{A}(\alpha)$ allows RRT to enjoy a low variance and hence requires very few samples in Monte-Carlo estimation.

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

$$\nabla_\alpha \widetilde{A}(\alpha) \approx \frac{\lambda}{N} \sum_{i=1}^{N} \nabla_\theta J(\theta, x)|_{\theta=g(\hat{\theta}_i;\alpha)} \tag{16}$$

where $\hat{\theta} \sim q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)$. The fact that g is differentiable *almost everywhere* with respect to α 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 $\hat{\theta}$ from $q\left(\hat{\theta}|\lfloor\alpha\rfloor_\Delta\right)$, namely, take $N = 1$ in (16).

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

On the discontinuities induced by RRT

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

To that end, we have the following result.

Lemma 2 *Suppose that J is ζ-lipschitz in θ and $A(\alpha)$ is γ-lipschitz in α. Then for any integer m,*

$$\left|\widetilde{A}(m\Delta) - \widetilde{A}(m\Delta - \epsilon)\right| < (\gamma + \zeta\lambda)\Delta$$

when $\epsilon \to \Delta$.

We note that when $\epsilon \to \Delta$, the quantity $\left|\widetilde{A}(m\Delta) - \widetilde{A}(m\Delta - \epsilon)\right|$ 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 Δ. On the other hand, in case one indeed chooses a relatively large Δ, 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, $\widetilde{A}(m\Delta) = A(m\Delta)$. And

$$\begin{aligned}
&\widetilde{A}(m\Delta - \epsilon) \\
=&\mathbb{E}_{q(\theta|(m-1)\Delta)} J(\theta + \lambda(\Delta - \epsilon)) \\
\approx&\mathbb{E}_{q(\theta|(m-1)\Delta)} \left\{ J(\theta) + \lambda(\Delta - \epsilon)J'(\theta) \right\} \\
=&A((m-1)\Delta) + \lambda(\Delta - \epsilon) \cdot \mathbb{E}_{q(\theta|(m-1)\Delta)} J'(\theta)
\end{aligned}$$

Since J is ζ-lipschitz,

$$\begin{aligned}
&A((m-1)\Delta) - \zeta\lambda(\Delta - \epsilon) \\
&< \widetilde{A}(m\Delta - \epsilon) < A((m-1)\Delta) + \zeta\lambda(\Delta - \epsilon)
\end{aligned}$$

It follows

$$\begin{aligned}
&A(m\Delta) - A((m-1)\Delta) + \zeta\lambda(\Delta - \epsilon) \\
&> \widetilde{A}(m\Delta) - \widetilde{A}(m\Delta - \epsilon) \\
&> A(m\Delta) - A((m-1)\Delta) - \zeta\lambda(\Delta - \epsilon)
\end{aligned}$$

Since $A(\cdot)$ is γ-lipschitz, then

$$\begin{aligned}
&\gamma\Delta + \zeta\lambda(\Delta - \epsilon) \\
&> \widetilde{A}(m\Delta) - \widetilde{A}(m\Delta - \epsilon) > -\gamma\Delta - \zeta\lambda(\Delta - \epsilon)
\end{aligned}$$

It follows

$$\left|\widetilde{A}(m\Delta) - \widetilde{A}(m\Delta - \epsilon)\right| < (\gamma + \zeta\lambda)\Delta$$

This proves the lemma. □

It is clear that when Δ 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, since the standard decoder appears to extract many repetitive topic words (see Appendix B.1).[2] On the synthetic datasets, we adopt the standard 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 30×500 topic-word probability matrix β_g is generated as the ground truth; each dataset is then generated based on β_g using different Dirichlet priors $\alpha_g \cdot \mathbf{1} \in \mathbb{R}^{30}$, where $\mathbf{1}$ denotes the all-one vector. We set α_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, [3] AGNews[4], DBPeida (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 β_g. The

[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 × 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 β matrix of a model. Note that the rows of T_L are arbitrarily ordered. To count how many words in the ith 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 λ 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.

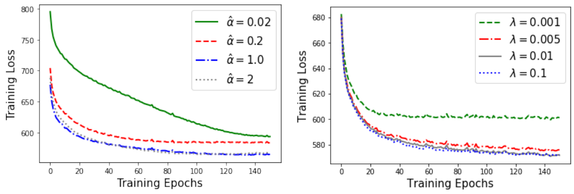


Figure 1: Training performance of RRT-VAE with different prior (left) and λ settings (right).

Prior Settings	0.02	0.2	1.0	2.0
Perplexity	1415	1130	951	875
NPMI	0.275	0.254	0.243	0.259
Sparsity	0.5353	0.1954	0.0868	0.0655

Table 2: Evaluation results on RRT-VAE with different prior settings. Perplexity: lower is better; NPMI: higher is better; Sparsity: higher means sparser.

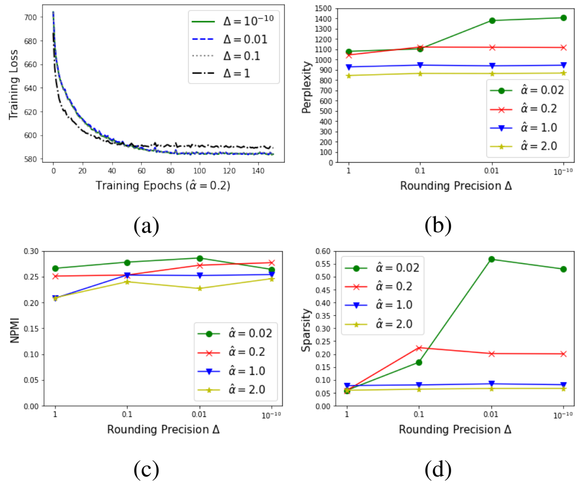
λ Settings	0.1	0.01	0.005	0.001
Perplexity	1004	951	978	1127
NPMI	0.221	0.243	0.271	0.160

Table 3: Evaluation results of RRT-VAE with different λ settings.

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 θ. The sparsity of θ 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 θ_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 θ, 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.

λ settings. The "gradient control" parameter λ 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 λ settings on the model performance, where the number of topics is set to 50 and the prior is set to 1. As shown, when λ 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 λ is set between around 0.01 and 0.005. Different λ settings can bring similar training performances but different testing results. For example, when λ 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.



(a)

(b)

(c)

(d)

Figure 2: (a) Training performance of RRT-VAE with different Δ settings; (b)-(d) perplexity, NPMI and sparsity of RRT-VAE with different Δ and prior $\hat{\alpha}$ settings. In these experiments, λ is set to 0.01, the number of topics is set to 50.

Influence of the rounding precision Δ. A main concern of RRT is that the induced discontinuities may cause training to be unstable. As proved in Section 3, this discontinuity actually causes little impact on the stability of training. We substantiate this conclusion in Figure 2 (a) by plotting the training loss curves of RRT-VAE under different Δ settings. As shown, all the training losses converge stably when using different Δ. This demonstrates that the precision of the rounding operation has little impact on the training stability. The influences of Δ on perplexity and NPMI are also modest. As shown in Figure 2 (b) and (c), the resulting perplexities and NPMIs are in general insensitive to the Δ settings.

From Figure 2 (b) and (d), it can also be observed that the perplexity of RRT-VAE has correlation with the sparsity. When Δ changes from 1 to 10^{-10}, the sparsity value of $\hat{\alpha} = 0.02$ (green line in Figure 2 (d)) jumps from 0.059 to around 0.55. [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 Δ 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 Δ setting hinders the loss converging to a lower value. Figure 3 (right) reports how different Δ settings influence the recovery accuracy of RRT-VAE on three synthetic datasets. It can be seen that a smaller Δ 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%.

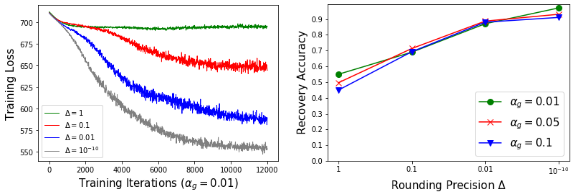


Figure 3: Training performances (left) and recovery accuracy (right) of RRT-VAE on a synthetic dataset ($\alpha_g = 0.01$) with different Δ settings.

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. A

[5] Since the sparsity we define is computed from the randomly sampled θ, it is inherently stochastic due to randomness in θ. 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 Δ 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 λ 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.

	20NG	AGNews	RCV1-v2	DBPedia	Yelp
50 topics	0.01	0.008	0.002	0.005	0.005
200 topics	0.005	0.005	0.002	0.003	0.003

Table 4: Optimal λ settings of RRT-VAE for different datasets.

	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	623/0.164	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 5: Perplexity/NPMI of the compared topic models on five datasets. The number of topic is set to 50.

	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.

margherita grimaldi pizzeria pepperoni sbarro brooklyn bianco mozza spinato concours udon ichiza monta tokyo chaya agedashi saigon chinatown gyoza yaki croissant decaf oatmeal scone coffe granola almond pastri latt muffin hue bo pho vietnames viet banh lemongrass vietnam mi basil sportsbook mandalay ronin kiki miyagi puck bachi shogun fatburg oxtail
heighten punctuat suppl amidst juxtapos conscious onward revel evok gleam ewwww saliva kneel cock toothless broom discust demerit surveil sill wan non asian pan asian pak taipei totti hotpot hai sift empty hand marshall stuffer overstock spree reorgan sweatshirt store preach outbreak heartfelt pois raymond uplift caregiv worship charismat deathli
buger haystack stripburg in and out quadrupl deli fukuburg fries food poison ambienc atmospher awsom bedienungen cafeteria defiantli chipotl slowest oldtown boozer after work carly grapevin fiver meet up hang tombston pokey pizza but peroni numero pizzaria pizza n nth insipid banal nil nla disposit st laurent hyper extraordinair procur
store sale housewar homegood inventori brows shelv thrift shopper stock sashimi eel tempura nigiri yellowtail ponzu sushi edamam tuna wasabi dr doctor exam physician nurs physician obgyn urgent clinic medic airport plane flight baggag mccarran tsa passeng megabu shuttl airlin workout instructor zumba yoga class bike gym crossfit fairway paintbal

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.

The experimental results are shown in Table 5 and 6. It can be seen that on the small and medium size datasets (20NG and AGNews), the performance of DirVAE levels with RRT-VAE, while on the large datasets (RCV1-v2, DBpedia and Yelp), the NPMI of RRT-VAE is significantly better than all the other compared models. Although the perplexity of NVDM is better than RRT-VAE, this gap is small. On the other hand, on NPMI, RRT-VAE outperforms NVDM by a very large margin. In fact, it has been demonstrated that perplexity is not necessarily a good metric for evaluating the quality of learned topics (Newman et al., 2010). Its correlation to the quality of the learned topics is questionable [6] (Chang et al., 2009). With these considerations, we argue that RRT-VAE is overall superior to other compared models.

Table 7 exhibits the extracted topic words of different models, where each line of the words corresponds to a certain topic. We see that the words extracted by RRT-VAE (the bottom cell of Table 7) are much more interpretable, from which it can

[6] 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 β matrix) and the learned encoder. Therefore it does not provide a direct evaluation of the learned word distributions in the β matrix. This problem is overcome by the additional NPMI measure, which is computed directly from the β matrix and serves as a more indicative quality measurement of the learned topics.

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
α_g=0.01	87.33%	84.0%	91.33%	**96.67%**
α_g=0.05	91.33%	83.0%	84.67%	**93.0%**
α_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 α_g settings. For RRT-VAE, λ is set to 1; Δ 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.

Francisco 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
α_g=0.01	50.33%	59.33%	61.33%
α_g=0.05	48.33%	64.67%	59.67%
α_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 α_g settings. The models adopt the same prod decoder structure. For RRT-VAE, λ is set to 1; Δ 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 Δ settings on two synthetic datasets ($\alpha = 0.05$ and $\alpha = 0.1$). The curves perform similarly to Figure 3 (left).

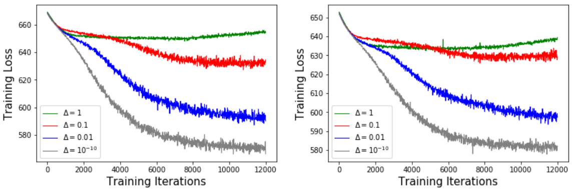


Figure 4: Training performances of RRT-VAE with different Δ settings. Left: $\alpha_g = 0.05$; right: $\alpha_g = 0.1$.

A.3 Recovered topic words

Table 10 exhibits an example of the ground truth topic word matrix T_g used in our experiments and the corresponding recovered matrix T_L learned by RRT-VAE. Note that the rows of T_L are arbitrarily ordered. The matching relation between T_L and T_g can be found using the evaluation method introduced in Section 5.2.

20 225 427 252 256 177 135 257 78 193	233 133 23 70 303 401 423 269 329 120
115 269 399 132 360 164 0 42 247 446	96 223 29 354 359 51 270 297 490 405
425 257 115 433 472 497 103 434 223 216	315 73 217 160 366 363 113 433 158 412
10 1 15 91 397 367 459 412 93 101	204 446 92 484 163 467 403 250 392 175
498 53 60 209 120 213 51 351 80 92	494 195 474 27 46 64 150 388 152 314
146 399 232 268 234 77 401 353 42 200	435 320 288 25 411 53 436 46 187 437
81 454 444 321 44 441 410 233 425 406	295 3 23 145 334 139 198 395 105 180
435 320 288 25 53 411 436 46 187 437	347 266 375 422 21 239 157 90 247 244
459 207 69 462 76 247 162 221 389 288	0 475 288 196 120 382 485 52 103 457
282 26 336 154 86 94 471 85 1 224	121 144 319 392 472 55 234 346 61 499
204 446 484 92 163 403 467 250 392 175	81 454 444 321 44 441 410 233 425 406
334 492 24 388 446 68 391 180 283 390	190 411 334 319 122 278 318 434 309 105
494 195 46 474 27 64 150 388 152 314	298 493 347 481 50 351 127 70 201 353
315 73 217 160 366 363 113 53 433 158	288 289 485 240 410 421 457 7 139 249
295 3 23 145 334 139 198 395 105 180	96 175 282 81 181 214 76 350 495 37
96 223 29 354 359 51 270 297 490 405	380 369 82 223 491 301 23 439 324 60
288 289 485 240 410 421 457 7 139 249	**20 225 427 252 177 256 135 257 78 193**
444 7 356 369 454 84 91 83 176 485	387 244 471 13 30 374 207 97 133 438
233 23 133 70 303 269 401 423 329 120	459 207 69 462 76 247 162 221 389 288
298 493 347 481 50 127 351 70 353 201	**115 269 399 132 360 164 0 42 247 213**
380 369 223 82 491 301 23 439 324 60	282 26 336 154 86 94 471 85 1 284
466 486 210 122 400 234 59 497 371 255	5 177 272 94 383 54 307 463 265 49
390 17 421 295 476 453 253 67 109 147	10 1 15 91 397 367 459 412 93 271
96 175 282 81 181 214 350 76 217 37	466 210 486 122 400 234 59 497 255 371
5 177 272 94 383 54 307 463 265 68	146 399 232 268 234 401 77 353 42 493
190 411 334 319 122 318 278 105 240 434	498 60 53 209 120 51 351 213 80 92
387 244 471 13 374 30 207 97 133 438	444 7 356 369 84 454 91 83 176 485
121 144 319 472 392 55 234 346 61 499	425 257 115 433 472 497 103 434 223 216
347 266 375 422 21 239 157 90 247 129	390 17 421 476 453 295 253 67 109 147
475 0 288 196 120 382 485 52 103 457	334 492 24 388 446 391 68 180 283 338

Table 10: Left: the ground truth topic word matrix T_g; Right: a matrix T_L learned by RRT-VAE. Note that the rows of T_L are arbitrarily ordered. For example, the first and second rows of T_g individually correspond to the 11th and 14th rows of T_L (as shown in bold).

B Additional Results on Real-world Datasets

B.1 Repetitive words

write article one get know like think say go use
write article get one know like use think say go
get go like write make people article insurance tax one
write article one get use like think know go say
know thanks please anyone write get email article post like

Table 11: The standard decoder appears to extract many repetitive words on 20NG.

As shown in Table 11, when using the standard decoder on the 20NG dataset, RRT-VAE appears to extract many repetitive topic words.

B.2 Performance of Online LDA on 20NG

	Perplexity	NPMI
50 topics	1183	0.181
200 topics	2728	0.162

Table 12: The experimental results of Online LDA on the 20NG dataset.

B.3 Topic words extracted by RRT-VAE

Table 13 exhibits the topic words extracted by RRT-VAE from four real-world datasets (20NG, AGNews, RCV1-v2 and DBpedia), where each line of the words corresponds to a certain topic.

health medical patient disease medicine estimate hospital care service coverage violent gun crime handgun usa criminal uk homicide defend firearm constitution senate amendment representative states president extend congress militia bear homosexual male sexual man statistics percent rsa number gay behavior fuel moon cool lunar air launch heat stage orbit cold
guilti conspiraci ghraib martha milosev enron prison yugoslav torture sentence ansari spaceshipon genesi space hubbl parachut spacecraft nasa station astronaut docomo nokia vodafon phone motorola blackberri ip mobil treo mmo kill explod injur dead quak typhoon peopl jakarta bomb landslide mice skeleton supercompute gene genetic stem clone ancestor scientist speci
thriv lifestyl shop museum flock fame cultur tast dream ancient desktop access network internet digit modem intranet download voice compute durum flood moisture disaster wheat grain hrw canol sorghum crop detain troop gunfire violent policeman military siege dozen terror embass attorney counsel felon lawsuit jury testif improp hear conspir guilt
paperback reprint book republish young adult isbn author locu scholast desktop server intel web bas software device microsoft applic uav clarinet bassist guitarist drummer banjo violin guitar drum saxophon keyboardist airway airport iata airlin icao brokerag telecommun exchang asset financi

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.