CS 6501 Natural Language Processing

Variational Auto-encoder

Yangfeng Ji

November 7, 2018

Department of Computer Science
University of Virginia

UNIVERSITY of VIRGINIA | ENGINEERING

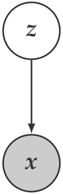
Overview

1. Variational Inference, Review
2. Variational Auto-encoder
3. Application: Neural variational document model

Variational Inference, Review

Latent Variable Models

$$p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = \underbrace{p(\boldsymbol{z}; \boldsymbol{\theta})}_{\text{prior}} \underbrace{p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})}_{\text{likelihood}} \tag{1}$$



Two Problems, One Connection

Two problems:

- What is $\boldsymbol{\theta}$ given a collection of data point $\{\boldsymbol{x}_n\}$?
- What is $p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})$?

One connection

- Computation of $p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})$

Ideal Cases

... where we can compute $p(\boldsymbol{x}; \boldsymbol{\theta})$, then

- We can get an estimate of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{2}$$

Ideal Cases

... where we can compute $p(\boldsymbol{x}; \boldsymbol{\theta})$, then

- We can get an estimate of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{2}$$

- Then, for a given data point $\boldsymbol{x}'$, we can compute its latent variable (representation) $\boldsymbol{z}$ as

$$p(\boldsymbol{z}' \mid \boldsymbol{x}'; \hat{\boldsymbol{\theta}}) = \frac{p(\boldsymbol{x}', \boldsymbol{z}'; \hat{\boldsymbol{\theta}})}{p(\boldsymbol{x}'; \hat{\boldsymbol{\theta}})} \tag{3}$$

Ideal Cases

... where we can compute $p(\boldsymbol{x}; \boldsymbol{\theta})$, then

- We can get an estimate of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{2}$$

- Then, for a given data point $\boldsymbol{x}'$, we can compute its latent variable (representation) $\boldsymbol{z}$ as

$$p(\boldsymbol{z}' \mid \boldsymbol{x}'; \hat{\boldsymbol{\theta}}) = \frac{p(\boldsymbol{x}', \boldsymbol{z}'; \hat{\boldsymbol{\theta}})}{p(\boldsymbol{x}'; \hat{\boldsymbol{\theta}})} \tag{3}$$

$p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})$ is challenging in practice

Variational Inference

- Approximate $p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})$ with a variational distribution $q(\boldsymbol{z}; \boldsymbol{\phi})$

$$q(\boldsymbol{z}; \boldsymbol{\theta}) \approx p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})) \tag{4}$$

Variational Inference

- Approximate $p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})$ with a variational distribution $q(\boldsymbol{z}; \boldsymbol{\phi})$

$$q(\boldsymbol{z}; \boldsymbol{\theta}) \approx p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})) \tag{4}$$

- Approximation by minimizing the KL divergence (still needs $p(\boldsymbol{x}; \boldsymbol{\theta})$, not directly solvable)

$$\mathrm{KL}(q \| p) = E_{q(\boldsymbol{z}; \boldsymbol{\phi})}[q(\boldsymbol{z}; \boldsymbol{\phi})] - E_{q(\boldsymbol{z}; \boldsymbol{\phi})}[p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})] \tag{5}$$

Variational Inference

- Approximate $p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})$ with a variational distribution $q(\boldsymbol{z}; \boldsymbol{\phi})$

$$q(\boldsymbol{z}; \boldsymbol{\theta}) \approx p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})) \tag{4}$$

- Approximation by minimizing the KL divergence (still needs $p(\boldsymbol{x}; \boldsymbol{\theta})$, not directly solvable)

$$\mathrm{KL}(q \| p) = E_{q(\boldsymbol{z}; \boldsymbol{\phi})}[q(\boldsymbol{z}; \boldsymbol{\phi})] - E_{q(\boldsymbol{z}; \boldsymbol{\phi})}[p(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\theta})] \tag{5}$$

- or, by maximizing the evidence lower bound (solvable with some constraints)

$$\mathrm{ELBo}(\boldsymbol{\theta}, \boldsymbol{\phi}) = E_q[p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})] - E_q[q(\boldsymbol{z}; \boldsymbol{\phi})] \tag{6}$$

Example: Mixture of Gaussians

- Prior:

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{7}$$

- Likelihood:

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \tag{8}$$

Example: Mixture of Gaussians

- Prior:

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{7}$$

- Likelihood:

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \tag{8}$$

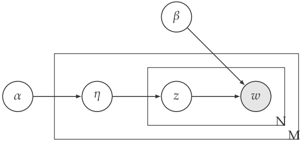
- $p(\boldsymbol{x})$ is tractable

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{9}$$

So, we can use the EM algorithm.

Example: Latent Dirichlet Allocation

1. Choose $\boldsymbol{\eta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
2. For each word $\boldsymbol{w}_n$
   2.1 Choose a topic $\boldsymbol{z}_n \sim$ Multinomial$(\boldsymbol{\eta})$
   2.2 Choose a word $\boldsymbol{w}_n \sim p(\boldsymbol{w}_n \mid \boldsymbol{z}_n; \boldsymbol{\beta})$, a multinomial probability conditioned on the topic $\boldsymbol{z}_n$.



Variational inference: mean-field approximation

Commen Setup

Choose your $p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})$ carefully

- Categorical (in GMMs)
- Multinomial (in LDA)
- Dirichlet (in LDA)

Question

What if $p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})$ is not a common probability distribution?

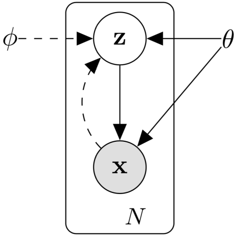
Variational Auto-encoder

Extension on Modeling

$$p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = p(\boldsymbol{z}; \boldsymbol{\theta}) \cdot p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta}). \tag{10}$$

- $p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})$ is modeled by a neural network with some randomness, therefore $E_q[p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})]$ is intractable
- variational distribution $q(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\phi})$

Graphical Representation



- $q(z \mid x; \phi)$: probabilistic encoder
- $p(x \mid z; \theta)$: probabilistic decoder

Evidence Lower Bound

$$
\begin{aligned}
\mathrm{ELBo}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= E_{q(z;\phi)}[\log p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})] - E_{q(z;\phi)}[\log q(\boldsymbol{z}; \boldsymbol{\phi})] \\
&= E_{q(z;\phi)}[\log p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})] + E_{q(z;\phi)}[p(\boldsymbol{z}; \boldsymbol{\theta})] \qquad (11) \\
&\quad -E_{q(z;\phi)}[\log q(\boldsymbol{z}; \boldsymbol{\phi})] \\
&= E_{q(z;\phi)}[\log p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})] - \mathrm{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\phi}) \| p(\boldsymbol{z}; \boldsymbol{\theta})) \qquad (12)
\end{aligned}
$$

Gradient based Learning

$$\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})] \quad = \quad \nabla_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})$$

1

Gradient based Learning

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})] &= \nabla_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})
\end{aligned}
$$

1

Gradient based Learning

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})] &= \nabla_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \frac{\nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi})}{q(\boldsymbol{z};\boldsymbol{\phi})} \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})
\end{aligned}
$$

1

Gradient based Learning

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})] &= \nabla_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \frac{\nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi})}{q(\boldsymbol{z};\boldsymbol{\phi})} \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})^1
\end{aligned}
$$

[1]likelihood ratio trick, also used in policy gradient

Gradient based Learning

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})] &= \nabla_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \frac{\nabla_{\boldsymbol{\phi}} q(\boldsymbol{z};\boldsymbol{\phi})}{q(\boldsymbol{z};\boldsymbol{\phi})} \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z};\boldsymbol{\phi}) \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z};\boldsymbol{\theta})^{1} \\
&\approx \frac{1}{L} \sum_{l=1}^{L} \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}^{(l)};\boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z}^{(l)};\boldsymbol{\theta})
\end{aligned}
$$

where $\boldsymbol{z}^{(l)} \sim q(\boldsymbol{z};\boldsymbol{\phi})$

[1]likelihood ratio trick, also used in policy gradient

Variance of MC Estimator

$$\nabla_{\boldsymbol{\phi}} E_{q(\boldsymbol{z};\boldsymbol{\phi})}[\log p(\boldsymbol{x} \mid \boldsymbol{z}; \boldsymbol{\theta})] \approx \frac{1}{L} \sum_{l=1}^{L} \nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{z}^{(l)}; \boldsymbol{\phi}) \log p(\boldsymbol{x} \mid \boldsymbol{z}^{(l)}; \boldsymbol{\theta}) \tag{13}$$

where $\boldsymbol{z}^{(l)} \sim q(\boldsymbol{z}; \boldsymbol{\phi})$.

Problems

- Sampling from $q(\boldsymbol{z}; \boldsymbol{\phi})$ is not easy
- Large variance

Reparameterization Trick

Representing $q(z; \phi)$ as a composition of a function g and a simple random variable ϵ

$$z = g_{\phi}(\epsilon, x) \tag{14}$$

Advantages

- g can be an arbitrary function (e.g., neural networks)
- ϵ is easier to sample

Reparameterization Trick: Example

- Original form $z \sim \mathcal{N}(\mu, \sigma^2)$
- Reparameterization form:
  - $z = \mu + \sigma\epsilon$
  - $\epsilon \sim \mathcal{N}(0, 1)$

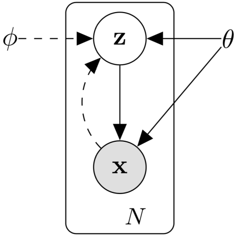
Approximation of ELBo

Based on the reparameterization trick, the ELBo can be approxiamted as

$$\tilde{\mathscr{L}} = \frac{1}{L}\sum_{l=1}^{L} \log p(\boldsymbol{x}_i \mid \boldsymbol{z}_i^{(l)}) - \mathrm{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\phi}) \| p(\boldsymbol{z}; \boldsymbol{\theta})) \quad (15)$$

where $\boldsymbol{z}_i^{(l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_i^{(l)}, \boldsymbol{x}_i)$ and $\boldsymbol{\epsilon}_i^{(l)} \sim p(\boldsymbol{\epsilon})$.

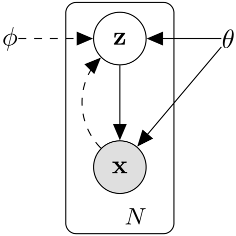
Comments



Encoder

- $\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- Neural networks with some simple randomness

Comments



Encoder

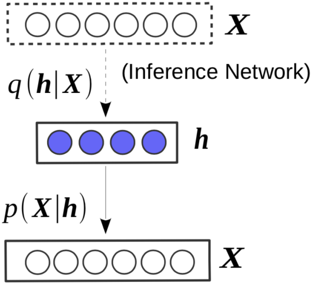
- $z = g_{\phi}(\epsilon, x)$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$
- Neural networks with some simple randomness

Decoder

- A simple nonlinear function

Application: Neural variational document model

Neural Variational Document Model



- an MLP document encoder $q(\boldsymbol{h} \mid \boldsymbol{x}; \boldsymbol{\theta})$
- a softmax decoder $p(\boldsymbol{x} \mid \boldsymbol{h}; \boldsymbol{\phi}) = \prod_i p(x_i \mid \boldsymbol{h}; \boldsymbol{\phi})$

[Miao et al., 2016]

Decoder

The decoder $p(x_i \mid \boldsymbol{h}$ is defined as

$$p(x_i \mid \boldsymbol{h}; \boldsymbol{\theta}) \propto \exp(\boldsymbol{h}^\top \mathbf{R} \boldsymbol{v}_{x_i} + \boldsymbol{b}_{x_i}) \qquad (16)$$

where $\mathbf{R}$ and $\{\boldsymbol{b}_{x_i}\}$ are part of $\boldsymbol{\theta}$, and $\boldsymbol{v}_{x_i}$ is the embedding of word x_i.

[Miao et al., 2016]

Encoder

The encoder $q(\boldsymbol{h} \mid \boldsymbol{x}; \boldsymbol{\phi})$ is defined as

$$q(\boldsymbol{h} \mid \boldsymbol{x}; \boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{h} \mid \boldsymbol{\mu}(\boldsymbol{x}), \operatorname{diag}(\boldsymbol{\sigma}^2(\boldsymbol{x}))) \tag{17}$$

with

$$\begin{aligned} \boldsymbol{\mu} &= l_1(\boldsymbol{\pi}) & (18) \\ \log \boldsymbol{\sigma} &= l_2(\boldsymbol{\pi}) & (19) \\ \boldsymbol{\pi} &= g(\boldsymbol{x}) & (20) \end{aligned}$$

where $g(\cdot)$ is a neural network and both l_1 and l_2 are linear transformations.

[Miao et al., 2016]

With the reparameterization trick, we have $\boldsymbol{h} \sim q(\boldsymbol{h} \mid \boldsymbol{x}; \boldsymbol{\phi})$

$$\boldsymbol{h} = \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon} \tag{21}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

ELBo

The problem can be solved by directly applying the algorithm proposed in [Kingma and Welling, 2014] to maximize the following ELBo (also, Eq. 5 in [Miao et al., 2016])

$$\mathscr{L} = E[\sum_i p(x_i \mid \boldsymbol{h}; \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{h} \mid \boldsymbol{x}; \boldsymbol{\phi}) \| p(\boldsymbol{h}; \boldsymbol{\theta}))] \tag{22}$$

where $p(\boldsymbol{h})$ is a Gaussian prior.

Perplexity

Model	Dim	20News	RCV1
LDA	50	1091	1437
LDA	200	1058	1142
RSM	50	953	988
docNADE	50	896	742
SBN	50	909	784
fDARN	50	917	724
fDARN	200	—-	598
NVDM	50	**836**	563
NVDM	200	852	**550**

[Miao et al., 2016]

Topics

Space	*Religion*	*Encryption*	*Sport*	*Policy*
orbit	muslims	rsa	goals	bush
lunar	worship	cryptography	pts	resources
solar	belief	crypto	teams	charles
shuttle	genocide	keys	league	austin
moon	jews	pgp	team	bill
launch	islam	license	players	resolution
fuel	christianity	secure	nhl	mr
nasa	atheists	key	stats	misc
satellite	muslim	escrow	min	piece
japanese	religious	trust	buf	marc

[Miao et al., 2016]

Summary

1. Variational Inference, Review
2. Variational Auto-encoder
3. Application: Neural variational document model

Reference

Kingma, D. P. and Welling, M. (2014).
Auto-encoding variational Bayes.
In *ICLR*.

Miao, Y., Yu, L., and Blunsom, P. (2016).
Neural variational inference for text processing.
In *International Conference on Machine Learning*, pages 1727–1736.