UPPSALA UNIVERSITET

Machine learning – Block 7

Måns Magnusson
Department of Statistics, Uppsala University

Autumn 2022

Assignment 6: Evaluation

- Some clarifications
- Still behind with grading

This week's lectures

- Variational autoencoders
- Probabilistic Topic Models

Section 2

Introduction

Why variational autoencoders and topic models?

- Popular approaches in industry and academia
- Probabilistic methods for unsupervised learning

UPPSALA UNIVERSITET

Why variational autoencoders and topic models?

- Popular approaches in industry and academia
- Probabilistic methods for unsupervised learning
- Aim of this lecture:
  - Describe the models
  - How to estimate these models
  - Explain what they are used for

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents
- Used for:
  - Identify "closeness" in high-dimensional data

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents

- Used for:
  - Identify "closeness" in high-dimensional data
  - Visualize/analyze data

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents
- Used for:
  - Identify "closeness" in high-dimensional data
  - Visualize/analyze data
  - Compression

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents
- Used for:
  - Identify "closeness" in high-dimensional data
  - Visualize/analyze data
  - Compression
  - Feature construction

Use Cases

- Variational autoencoders: Unsupervised modeling of images
- Topic models: Unsupervised modeling of documents

- Used for:
  - Identify "closeness" in high-dimensional data
  - Visualize/analyze data
  - Compression
  - Feature construction
  - Analyze underlying patterns

Use Cases: Examples

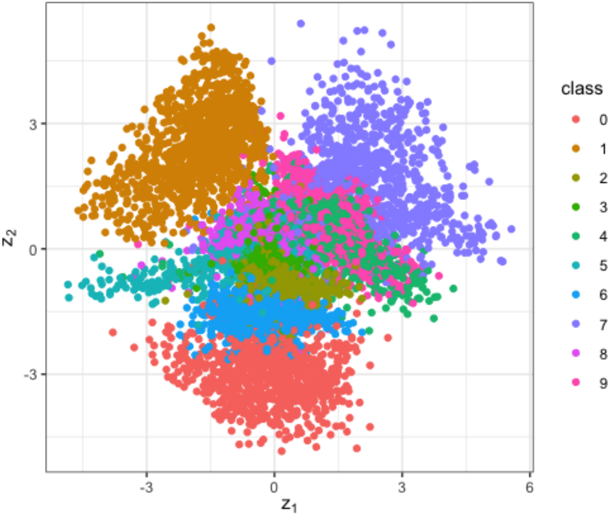


Figure: The latent state of MNIST using an Variational Autoencoder

Section 3

Autoencoders

Autoencoder

- An autoencoder is a neural network (e.g. feed-forward) that take an input x and predict (the same) x (r).

Autoencoder

- An autoencoder is a neural network (e.g. feed-forward) that take an input x and predict (the same) x (r).
- Three parts:
  - encoder $f(x)$ (or $e(x)$)
  - code h
  - decoder $g(h)$ (or $d(z)$)

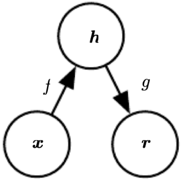


Figure: A Neural Autoencoder (Goodfellow et al, 2018)

Autoencoder

- An autoencoder is a neural network (e.g. feed-forward) that take an input x and predict (the same) x (r).
- Three parts:
  - encoder $f(x)$ (or $e(x)$)
  - code h
  - decoder $g(h)$ (or $d(z)$)

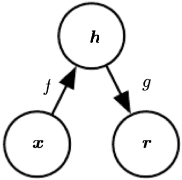


Figure: A Neural Autoencoder (Goodfellow et al, 2018)

- Loss function (reconstruction error):

$$L(\phi, \theta) = (x - d_\phi(e_\theta(x)))^2$$

The Undercomplete Autoencoder

- More interesting: an undercomplete autoencoder: Dimension of code is lower than that of x

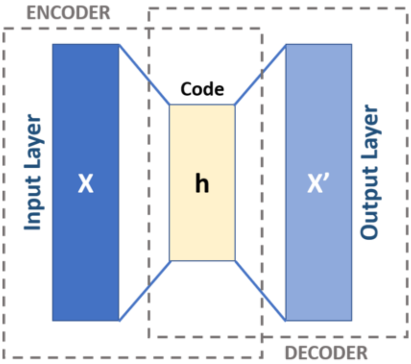


Figure: A Neural Autoencoder (Wikipedia)

PCA and autoencoders

- A linear autoencoder: $e_\theta(x) = W_\phi$, and $d_\theta(x) = W_\phi$
- We want to minimize the loss (ignoring b/the mean):

$$L(\phi, \theta) = \sum_{i=1}^{N} (x_i - W_\theta W_\phi x_i)^2$$

UPPSALA UNIVERSITET

PCA and autoencoders

- A linear autoencoder: $e_\theta(x) = W_\phi$, and $d_\theta(x) = W_\phi$
- We want to minimize the loss (ignoring b/the mean):

$$L(\phi, \theta) = \sum_{i=1}^{N} (x_i - W_\theta W_\phi x_i)^2$$

- Remember PCA loss:

$$L(P) = \sum_{i=1}^{N} (x_i - P_q P_q^T x_i)^2 \,,$$

where P is an orthogonal matrix of rank q.

UPPSALA UNIVERSITET

PCA and autoencoders

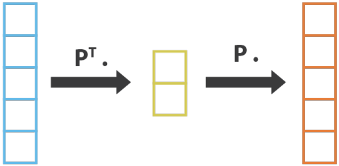
- A linear autoencoder: $e_\theta(x) = W_\phi$, and $d_\theta(x) = W_\phi$
- We want to minimize the loss (ignoring b/the mean):

$$L(\phi, \theta) = \sum_{i=1}^{N} (x_i - W_\theta W_\phi x_i)^2$$

- Remember PCA loss:

$$L(P) = \sum_{i=1}^{N} (x_i - P_q P_q^T x_i)^2 ,$$

where P is an orthogonal matrix of rank q.
- Hence: PCA can be seen as an autoencoder



Deep Autoencoders

- Deep Autoencoder: An autoencoder with multilayer neural networks as encoder and decoder
  - can be seen as a non-linear PCA
  - learn nonlinear representations

Deep Autoencoders

- Deep Autoencoder: An autoencoder with multilayer neural networks as encoder and decoder
  - can be seen as a non-linear PCA
  - learn nonlinear representations
- Problem: Deep autoencoders needs to be regularized to not overfit the latent state

UPPSALA
UNIVERSITET

probabilistic PCA as an decoder

- Problem: Autoencoders (as PCA) are not probabilistic models:
  - cannot generate data.
  - no notion of uncertainty

probabilistic PCA as an decoder

- Problem: Autoencoders (as PCA) are not probabilistic models:
  - cannot generate data.
  - no notion of uncertainty
- We would like something like probabilistic PCA for (deep) autoencoders

probabilistic PCA as an decoder

- Remember the pPCA model (with z as latent variable):

$$x_i \sim N(b + W z_i^T, \sigma I)$$

probabilistic PCA as an decoder

- Remember the pPCA model (with z as latent variable):

$$x_i \sim N(b + W z_i^T, \sigma I)$$

- Now, swap the simple parameters with a neural network

$$x_i \sim N(\text{NeuralNetwork}_\phi(z_i), \sigma I)$$

UPPSALA UNIVERSITET

probabilistic PCA as an decoder

- Remember the pPCA model (with z as latent variable):

$$x_i \sim N(b + W z_i^T, \sigma I)$$

- Now, swap the simple parameters with a neural network

$$x_i \sim N(\text{NeuralNetwork}_\phi(z_i), \sigma I)$$

- This is an example of a Deep Latent Variable model (a probabilistic decoder)

probabilistic PCA as an decoder

- Remember the pPCA model (with z as latent variable):

$$x_i \sim N(b + W z_i^T, \sigma I)$$

- Now, swap the simple parameters with a neural network

$$x_i \sim N(\text{NeuralNetwork}_\phi(z_i), \sigma I)$$

- This is an example of a Deep Latent Variable model (a probabilistic decoder)
- Another example is the Variational Autoencoder

Section 4

The Variational Autoencoder

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images
- Consists of:
  1. The (probabilistic) encoder $q(z|\phi, x)$: inference model

UPPSALA UNIVERSITET

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images
- Consists of:
  1. The (probabilistic) encoder $q(z|\phi, x)$: inference model
  2. Sample z from encoded x: the latent state

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images
- Consists of:
  1. The (probabilistic) encoder $q(z|\phi, x)$: inference model
  2. Sample z from encoded x: the latent state
  3. The (probabilistic) decoder $p(x|\theta, z)$: observation model

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images
- Consists of:
  1. The (probabilistic) encoder $q(z|\phi, x)$: inference model
  2. Sample z from encoded x: the latent state
  3. The (probabilistic) decoder $p(x|\theta, z)$: observation model
- Encoding the latent state as a distribution forces the space to be "reasonable"/reduces overfitting

The Variational Autoencoder (VAE)

- The variational autoencoder (VAE) is a deep probabilistic autoencoder
- Used for unsupervised learning of images
- Consists of:
  1. The (probabilistic) encoder $q(z|\phi, x)$: inference model
  2. Sample z from encoded x: the latent state
  3. The (probabilistic) decoder $p(x|\theta, z)$: observation model
- Encoding the latent state as a distribution forces the space to be "reasonable"/reduces overfitting
- VAEs get their name from variational inference (used in training)

The Variational Autoencoder

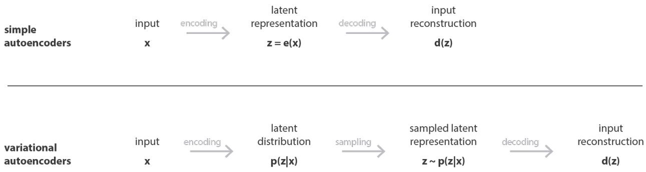


Figure: Autoencoder vs. the Variational Autoencoder (Rocca, 2019)

The Variational Autoencoder

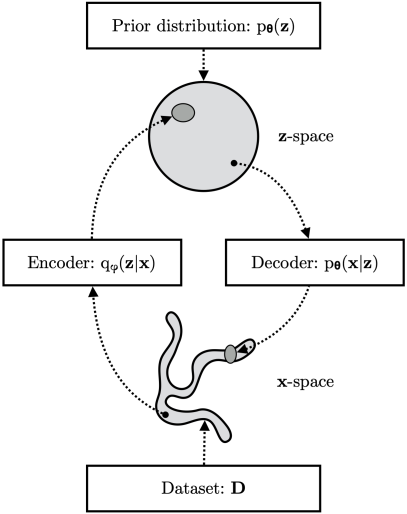


Figure: The Variational Autoencoder (Kingma and Welling, 2018, Fig. 2.1)

UPPSALA UNIVERSITET

The probabilistic decoder

- The probabilistic decoder $p(x|\theta, z)$ (observation model)
- Usually a Normal distribution:

$$x_i \sim N(\text{NeuralNetwork}(z, \theta), cI)$$

- x_i for observation i depends non-linearly on the latent state z_i

UPPSALA UNIVERSITET

The probabilistic decoder

- The probabilistic decoder $p(x|\theta, z)$ (observation model)
- Usually a Normal distribution:

$$x_i \sim N(\text{NeuralNetwork}(z, \theta), cI)$$

- x_i for observation i depends non-linearly on the latent state z_i
- A probabilistic linear decoder: pPCA

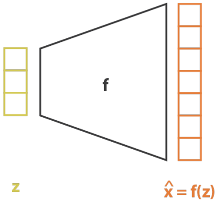


Figure: The Decoder (Rocca, 2019)

The probabilistic encoder

- The probabilistic encoder $q(z|x, \phi)$ (inference model)

The probabilistic encoder

- The probabilistic encoder $q(z|x, \phi)$ (inference model)
- We assume that $q_\phi(z|x)$ follows a specific distribution. Commonly:

$$z \sim N(\mu, \Sigma)$$

The probabilistic encoder

- The probabilistic encoder $q(z|x, \phi)$ (inference model)
- We assume that $q_\phi(z|x)$ follows a specific distribution. Commonly:

$$z \sim N(\mu, \Sigma)$$

- A neural network learns the parameters μ and Σ

$$\mu = \mathrm{NeuralNetwork}(x, \phi_\mu)\,,$$

and

$$\Sigma = \mathrm{NeuralNetwork}(x, \phi_\Sigma)\,.$$

UPPSALA UNIVERSITET

UPPSALA UNIVERSITET

The probabilistic encoder

- The probabilistic encoder $q(z|x, \phi)$ (inference model)
- We assume that $q_\phi(z|x)$ follows a specific distribution. Commonly:

$$z \sim N(\mu, \Sigma)$$

- A neural network learns the parameters μ and Σ

$$\mu = \text{NeuralNetwork}(x, \phi_\mu)\,,$$

and

$$\Sigma = \text{NeuralNetwork}(x, \phi_\Sigma)\,.$$

- Common assumption: Σ is a diagonal matrix.

The probabilistic encoder

- The probabilistic encoder $q(z|x,\phi)$ (inference model)
- We assume that $q_\phi(z|x)$ follows a specific distribution. Commonly:

$$z \sim N(\mu, \Sigma)$$

- A neural network learns the parameters μ and Σ

$$\mu = \mathrm{NeuralNetwork}(x, \phi_\mu)\,,$$

and

$$\Sigma = \mathrm{NeuralNetwork}(x, \phi_\Sigma)\,.$$

- Common assumption: Σ is a diagonal matrix.
- Result: z_i for observation i depends non-linearly on x_i

The probabilistic encoder

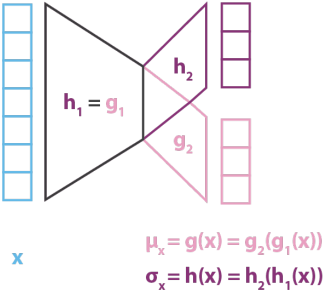


Figure: The Encoder (Rocca, 2019)

The Variational Autoencoder

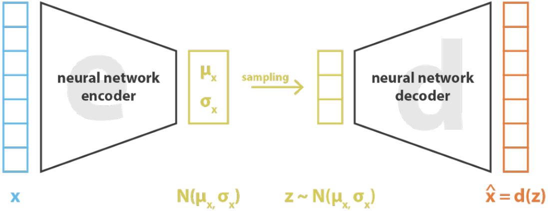


Figure: The Variational Autoencoder (Rocca, 2019)

UPPSALA UNIVERSITET

The Variational Autoencoder

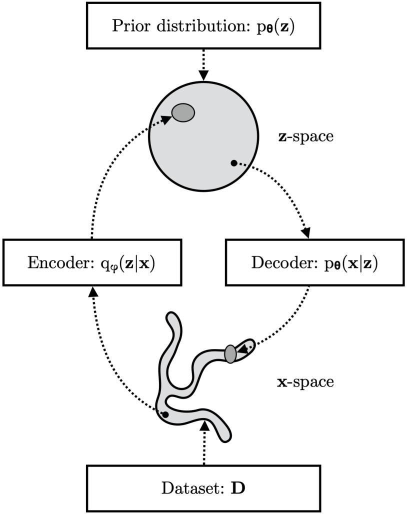


Figure: The Variational Autoencoder (Kingma and Welling, 2018, Fig. 2.1)

Training a VAE

- Goal: estimating ϕ, θ (and z_i)

Training a VAE

- Goal: estimating ϕ, θ (and z_i)
- The encoder and decoder are (usually) complicated (no close form solution)
- Estimate ϕ and θ using gradient ascent

Training a VAE

- Goal: estimating ϕ, θ (and z_i)
- The encoder and decoder are (usually) complicated (no close form solution)
- Estimate ϕ and θ using gradient ascent
- Target:
  - Maximize $\log p(x)$
    (Explain the data as well as possible)

UPPSALA UNIVERSITET

Training a VAE

- Goal: estimating ϕ, θ (and z_i)
- The encoder and decoder are (usually) complicated (no close form solution)
- Estimate ϕ and θ using gradient ascent
- Target:
  - Maximize $\log p(x)$
    (Explain the data as well as possible)
- Optimization target:
  Maximize the variational lower bound or evidence lower bound (ELBO)

The marginal log-likelihood

$$
\begin{aligned}
\log p_\theta(x) &= \int q_\phi(z|x) \log p_\theta(x) dz \\
&= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{p_\theta(z|x)}\right)\right], \text{using } p(z|x) = \frac{p(x,z)}{p(x)} \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] \\
&= \underbrace{\mathcal{L}_{\theta,\phi}(x)}_{\text{ELBO}} + D_{KL}(q_\phi(z|x)||p_\theta(z|x))
\end{aligned}
$$

The marginal log-likelihood

$$\begin{aligned}
\log p_\theta(x) &= \int q_\phi(z|x) \log p_\theta(x) dz \\
&= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{p_\theta(z|x)}\right)\right], \text{using } p(z|x) = \frac{p(x,z)}{p(x)} \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] \\
&= \underbrace{\mathcal{L}_{\theta,\phi}(x)}_{\text{ELBO}} + D_{KL}(q_\phi(z|x)||p_\theta(z|x))
\end{aligned}$$

$$\underbrace{\mathcal{L}_{\theta,\phi}(x)}_{\text{ELBO}} = \log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

The Kullback-Leibler divergence

- The Kulback-Leibler divergence: a way of measuring the "distance" between probability distributions

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x)) = \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right]$$

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \geq 0$$

Training target

- Optimization target: Maximize the ELBO

$$\mathcal{L}_{\theta,\phi}(x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

- ELBO is a lower bound for the marginal log-likelihood (similar to the EM algorithm)

Training target

- Optimization target: Maximize the ELBO

$$\mathcal{L}_{\theta,\phi}(x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

- ELBO is a lower bound for the marginal log-likelihood (similar to the EM algorithm)
- Maximizing the ELBO will do two things:
  - Maximize the marginal log-likelihood $\log p_\theta(x)$: Better generative model/decoder
  - Minimize the KL-divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$: Better approximation of the latent space/encoder

Optimizing the ELBO

- Stochastic Gradient *Ascent* to maximize:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) &= \sum_{i}^{N} \mathcal{L}_{\theta,\phi}(x_i) \\ &= \sum_{i}^{N} \mathbb{E}_{q_\phi(z_i|x_i)} \left[\log\left(p_\theta(x_i, z_i)\right) - \log(q_\phi(z_i|x_i))\right]\end{aligned}$$

Optimizing the ELBO

- Stochastic Gradient *Ascent* to maximize:

$$
\begin{aligned}
\mathcal{L}_{\theta,\phi}(x) &= \sum_{i}^{N} \mathcal{L}_{\theta,\phi}(x_i) \\
&= \sum_{i}^{N} \mathbb{E}_{q_\phi(z_i|x_i)} \left[\log\left(p_\theta(x_i, z_i)\right) - \log(q_\phi(z_i|x_i))\right]
\end{aligned}
$$

- Two problems:
  1. How do we compute the expectation?
     Solution: Monte Carlo Approximation

Optimizing the ELBO

- Stochastic Gradient *Ascent* to maximize:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) &= \sum_{i}^{N} \mathcal{L}_{\theta,\phi}(x_i) \\ &= \sum_{i}^{N} \mathbb{E}_{q_\phi(z_i|x_i)} \left[\log\left(p_\theta(x_i, z_i)\right) - \log(q_\phi(z_i|x_i))\right]\end{aligned}$$

- Two problems:
  1. How do we compute the expectation?
     Solution: Monte Carlo Approximation
  2. How compute the gradient wrt ϕ?
     Solution: Change of variables: $z = g(\epsilon, \phi, x)$
     This is called the reparametrization trick

Optimizing the ELBO

- Using the reparametrization trick and Monte Carlo approximation, we get:

$$\begin{aligned}
\mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(p_\theta(x,z)\right) - \log(q_\phi(z|x))\right] \\
&= \mathbb{E}_{p(\epsilon)}\left[\log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\right] \\
&\approx \log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))
\end{aligned}$$

Optimizing the ELBO

- Using the reparametrization trick and Monte Carlo approximation, we get:

$$
\begin{aligned}
\mathcal{L}_{\theta,\phi}(x) =& \mathbb{E}_{q_\phi(z|x)}\left[\log\left(p_\theta(x,z)\right) - \log(q_\phi(z|x))\right] \\
=& \mathbb{E}_{p(\epsilon)}\left[\log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\right] \\
\approx& \log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))
\end{aligned}
$$

- A common approach: do the MC approximation with only one sample per datapoint x_i.

Optimizing the ELBO

UPPSALA UNIVERSITET

- Using the reparametrization trick and Monte Carlo approximation, we get:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) =& \mathbb{E}_{q_\phi(z|x)}\left[\log\left(p_\theta(x,z)\right) - \log(q_\phi(z|x))\right] \\ =& \mathbb{E}_{p(\epsilon)}\left[\log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\right] \\ \approx& \log\left(p_\theta(x, g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\end{aligned}$$

- A common approach: do the MC approximation with only one sample per datapoint x_i.
- We approximate both $\mathcal{L}_{\theta,\phi}(x)$ and $\nabla\mathcal{L}_{\theta,\phi}(x)$

Optimizing the ELBO

- Using the reparametrization trick and Monte Carlo approximation, we get:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) =&\mathbb{E}_{q_\phi(z|x)}\left[\log\left(p_\theta(x,z)\right) - \log(q_\phi(z|x))\right]\\ =&\mathbb{E}_{p(\epsilon)}\left[\log\left(p_\theta(x,g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\right]\\ \approx&\log\left(p_\theta(x,g(\epsilon,\phi,x))\right) - \log(q_\phi(g(\epsilon,\phi,x)|x))\end{aligned}$$

- A common approach: do the MC approximation with only one sample per datapoint x_i.
- We approximate both $\mathcal{L}_{\theta,\phi}(x)$ and $\nabla\mathcal{L}_{\theta,\phi}(x)$
- Sometimes called a doubly stochastic algorithm.

The Autoencoding Variational Bayes Algorithm

Algorithm 1: Stochastic optimization of the ELBO. Since noise originates from both the minibatch sampling and sampling of $p(\boldsymbol{\epsilon})$, this is a doubly stochastic optimization procedure. We also refer to this procedure as the *Auto-Encoding Variational Bayes* (AEVB) algorithm.

Data:
  $\mathcal{D}$: Dataset
  $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$: Inference model
  $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$: Generative model
Result:
  $\boldsymbol{\theta}, \boldsymbol{\phi}$: Learned parameters

$(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow$ Initialize parameters
while *SGD not converged* **do**
  $\mathcal{M} \sim \mathcal{D}$ (Random minibatch of data)
  $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ (Random noise for every datapoint in $\mathcal{M}$)
  Compute $\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{M}, \boldsymbol{\epsilon})$ and its gradients $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{M}, \boldsymbol{\epsilon})$
  Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using SGD optimizer
end

Figure: The Autoencoding Variational Bayes Algorithm (Kingma and Welling, 2018, Algo. 1)

The Autoencoding Variational Bayes Algorithm

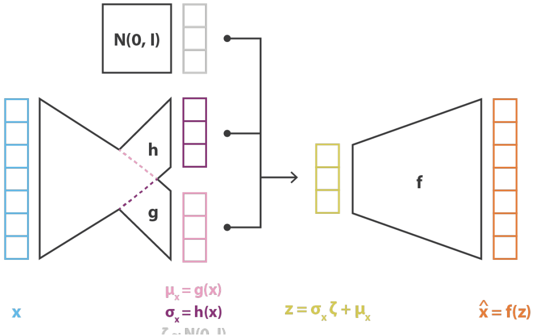


Figure: The Autoencoding Variational Bayes Algorithm (Rocca, 2019)

Summary

- Benefits of VAE:
  - Get a more interpretable latent state
  - We can estimate uncertainty
  - Regularize the latent state
  - We can inject knowledge in our latent state

Summary

- Benefits of VAE:
  - Get a more interpretable latent state
  - We can estimate uncertainty
  - Regularize the latent state
  - We can inject knowledge in our latent state
- Problems:
  - The blurry image problem

Summary

- Benefits of VAE:
  - Get a more interpretable latent state
  - We can estimate uncertainty
  - Regularize the latent state
  - We can inject knowledge in our latent state
- Problems:
  - The blurry image problem
- Still much ongoing research:

Figure: Examples of images generated with a deep hierarchical Variational Autoencoder (Vahdat and Kautz, 2020)

UPPSALA UNIVERSITET

Section 5

Probabilistic Topic Models

Probabilistic Topic Models

- Unsupervised method for textual data

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis
- Use cases:
  - Create features for supervised models

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis
- Use cases:
  - Create features for supervised models
  - Integrated in neural networks for model efficient learning (?)

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis
- Use cases:
  - Create features for supervised models
  - Integrated in neural networks for model efficient learning (?)
  - Visualize document collections

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis
- Use cases:
  - Create features for supervised models
  - Integrated in neural networks for model efficient learning (?)
  - Visualize document collections
  - Analyzing large corpora using statistical methods

Probabilistic Topic Models

- Unsupervised method for textual data
- Popular in industry and academia to analyze large corpora
- The most common model: Latent Dirichlet Allocation
- A mixed membership model (a mixture of multinomial mixtures model)
- Topic model builds on the the distributional hypothesis
- Use cases:
  - Create features for supervised models
  - Integrated in neural networks for model efficient learning (?)
  - Visualize document collections
  - Analyzing large corpora using statistical methods
- Example: All ears media monitoring of speech data

The Dirichlet Distribution

- Probability distribution over the simplex with K categories:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

and where

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$$

The Dirichlet Distribution

- Probability distribution over the simplex with K categories:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

and where

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$$

- The probability distribution has the support on the simplex, that is

$$\sum_{i=1}^{K} x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i \in [1, K]$$

The Dirichlet Distribution

- Probability distribution over the simplex with K categories:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

and where

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$$

- The probability distribution has the support on the simplex, that is

$$\sum_{i=1}^{K} x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i \in [1, K]$$

- The parameters $\boldsymbol{\alpha}$ can be seen as pseudo-counts

The Dirichlet Distribution

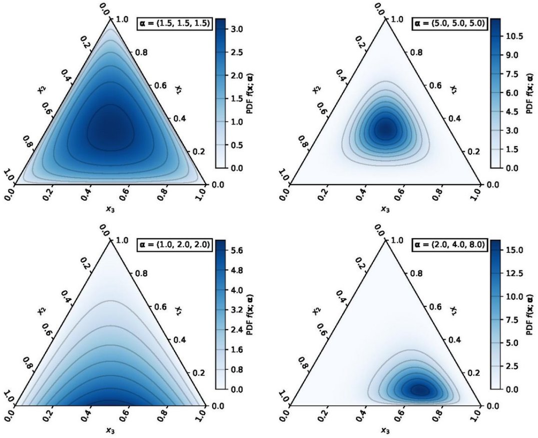


Figure: The Dirichlet Distribution (Wikipedia)

Interactive plot here.

UPPSALA UNIVERSITET

The distributional hypothesis

- Harris (1954) and Firths (1957):
  "Word is characterized by the company it keeps"

The distributional hypothesis

- Harris (1954) and Firths (1957):
  "Word is characterized by the company it keeps"
- Semantics (broadly defined) is captured by context

The distributional hypothesis

- Harris (1954) and Firths (1957):
  "Word is characterized by the company it keeps"
- Semantics (broadly defined) is captured by context
- Rough definition: word windows of different sizes

The distributional hypothesis

- Harris (1954) and Firths (1957):
  "Word is characterized by the company it keeps"
- Semantics (broadly defined) is captured by context
- Rough definition: word windows of different sizes
- Different window sizes, different semantic content:
  - Word embeddings (context: word windows)
  - Topic models (context: documents)

Example

1. "A friend in need is a friend indeed."
2. "She is my friend indeed."

UPPSALA UNIVERSITET

Latent Dirichlet Allocation

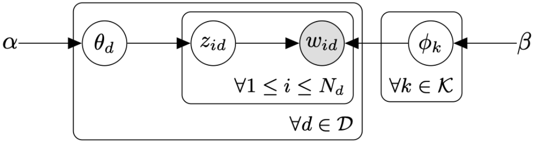


Figure: The Latent Dirichlet Allocation Model

where ϕ_k is the kth row in Φ (of dimension $K \times V$) and θ_d is the dth row in Θ (of dimension $D \times K$).

Generative model for LDA

Relies on the bag-of-word assumption

1. For each component k to K:
   1.1 $\phi_k \sim \text{Dirichlet}(\beta)$
2. For each document d:
   2.1 $\theta_d \sim \text{Dirichlet}(\alpha)$
   2.2 For each token i:
      2.2.1 $z_{id} \sim \text{Categorical}(\theta_d)$
      2.2.2 $w_{id} \sim \text{Categorical}(\phi_{z_{id}})$

Example of parameters $\mathbf{z}$, Θ and Φ

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

Example of parameters $\mathbf{z}$, Θ and Φ

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

$\Phi =$

	boat	shore	soccer	Zlatan	bank	money
Topic 1	0.35	0.35	0.05	0.05	0.15	0.05
Topic 2	0.025	0.025	0.45	0.45	0.025	0.025
Topic 3	0.025	0.025	0.025	0.025	0.45	0.45

Example of parameters $\mathbf{z}$, Θ and Φ

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

$\Phi =$

	boat	shore	soccer	Zlatan	bank	money
Topic 1	0.35	0.35	0.05	0.05	0.15	0.05
Topic 2	0.025	0.025	0.45	0.45	0.025	0.025
Topic 3	0.025	0.025	0.025	0.025	0.45	0.45

$\Theta =$

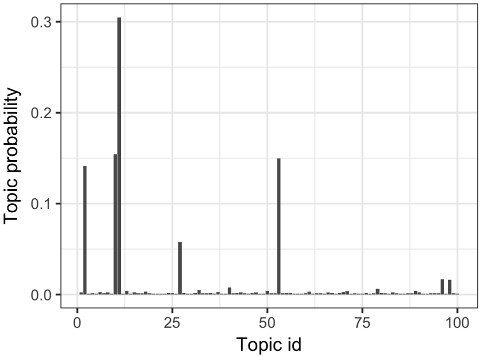
	Topic 1	Topic 2	Topic 3
doc 1	0.96	0.02	0.02
doc 2	0.3	0.2	0.5
doc 3	0.05	0.35	0.6

Closing arguments were heard yesterday in the Federal bankruptcy fraud trial of Stephen J. Sabbeth, whose legal problems have raised doubts about his ability to continue as leader of the Nassau County Democratic Party.

Mr. Sabbeth is charged with trying to conceal $750,000 from his bank creditors by hiding the money in a secret account in his wife's maiden name, rather than use it to pay creditors when his lumber business went into bankruptcy 10 years ago.

– The New York Times 25th of Febuary 1999

The estimated topic proportion ($\hat{\theta}_d$)



Topic top words

Topic	Top words (by ϕ_{kv})
2	party election voters campaign democratic
10	bank banks loans loan insurance savings
11	trial prison jury prosecutors convicted guilty
53	investigation inquiry documents investigators

Table: The words with highest probability ($p(w|k)$) for topic 2, 10, 11 and 53.

The Latent Dirichlet Allocation Model

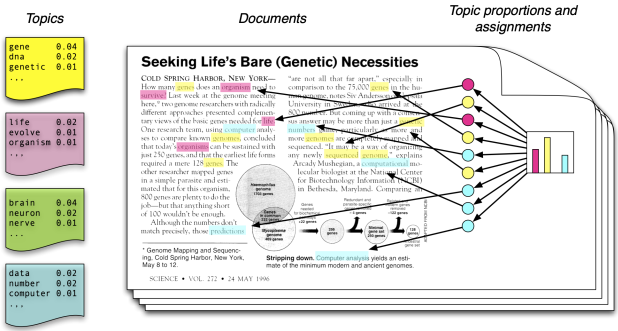


Figure: The Latent Dirichlet Allocation Model (Blei 2012, Fig. 1)

Inference

- Common inference approaches
  1. Variational inference
  2. Markov Chain Monte Carlo (MCMC)

Inference

- Common inference approaches
  1. Variational inference
  2. Markov Chain Monte Carlo (MCMC)
- The Gibbs sampler is usually prefered

Inference

- Common inference approaches
  1. Variational inference
  2. Markov Chain Monte Carlo (MCMC)
- The Gibbs sampler is usually prefered
- Similar to (Stochastic) EM

Gibbs sampler for LDA

The basic Gibbs sampler:

1. We want to estimate z, Φ, Θ:

Gibbs sampler for LDA

The basic Gibbs sampler:

1. We want to estimate z, Φ, Θ:
2. Sample topic indicators (latent variable)

$$p(z = k|\Phi, \Theta) \propto \phi_{v,k}\theta_{k,d}$$

Gibbs sampler for LDA

The basic Gibbs sampler:

1. We want to estimate z, Φ, Θ:
2. Sample topic indicators (latent variable)

$$p(z = k|\Phi, \Theta) \propto \phi_{v,k}\theta_{k,d}$$

3. Sample model parameters

$$\theta_d|\mathbf{z} \sim Dir(\mathbf{n}^{(d)} + \alpha)$$

$$\phi_k|\mathbf{z} \sim Dir(\mathbf{n}^{(v)} + \beta)$$

where $\mathbf{n}^{(d)}$ is the number of tokens by topic in document d and $\mathbf{n}^{(v)}$ is the number of tokens by topic for word type v.

Gibbs sampler for LDA

Integrating out (collapsing) Θ and Φ

$$p(\mathbf{z}|\mathbf{w}) = \int\int p(\mathbf{z},\Theta,\Phi|\mathbf{w}) \cdot p(\mathbf{z},\Theta,\Phi) d\Phi d\Theta$$

will result in the following Gibbs sampler:

$$p(z_i = k | w_i, \mathbf{z}_{\neg i}) \propto \underbrace{\frac{n_k^{(v)} + \beta}{n_k^{(v)} + V\beta}}_{type-topic\ (\Phi)} \cdot \underbrace{(n_k^{(d)} + \alpha)}_{topic-doc\ (\Theta)}$$

where $n^{(v)}$ and $n^{(d)}$ are count matrices of size $D \times K$ and $K \times V$.

Example of $n^{(v)}$ and $n^{(d)}$

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

Example of $n^{(v)}$ and $n^{(d)}$

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

$$n^{(v)} = \begin{array}{cccccc} \text{boat} & \text{shore} & \text{soccer} & \text{Zlatan} & \text{bank} & \text{money} \\ 2 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{array}$$

Example of $n^{(v)}$ and $n^{(d)}$

$\mathbf{w}_1$	boat	shore	bank		
$\mathbf{z}_1$	1	1	1		
$\mathbf{w}_2$	Zlatan	boat	shore	money	bank
$\mathbf{z}_2$	2	1	1	3	3
$\mathbf{w}_3$	money	bank	soccer	money	
$\mathbf{z}_3$	3	3	2	3	

$$n^{(v)} = \begin{array}{cccccc} \text{boat} & \text{shore} & \text{soccer} & \text{Zlatan} & \text{bank} & \text{money} \\ 2 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{array}$$

$$n^{(d)} = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 3 \\ 0 & 2 & 3 \end{bmatrix}$$

Topic Models as non-negative matrix factorization

$$\begin{bmatrix} \\ n_{dv} \\ (D \times V) \\ \\ \end{bmatrix} \approx \begin{bmatrix} \\ \Theta \\ (D \times K) \\ \\ \end{bmatrix} \times \begin{bmatrix} \Phi \\ (K \times V \end{bmatrix}$$

UPPSALA UNIVERSITET

- Practicalities
- Introduction
- Autoencoders
- The Variational Autoencoder
  - The probabilistic decoder
  - The encoder
  - Traing a variational autoencoder
- Probabilistic Topic Models
  - Latent Dirichlet Allocation
  - Estimating the LDA model

UPPSALA UNIVERSITET

Practicalities

- Setting K, α and β

Practicalities

- Setting K, α and β
- Reducing the vocabulary: stopwords, rare words, stemming

Practicalities

- Setting K, α and β
- Reducing the vocabulary: stopwords, rare words, stemming
- "Junk" topics

Practicalities

- Setting K, α and β
- Reducing the vocabulary: stopwords, rare words, stemming
- "Junk" topics
- We can analyze the topic indicators z directly

UPPSALA UNIVERSITET

- Practicalities
- Introduction
- Autoencoders
- The Variational Autoencoder
  - The probabilistic decoder
  - The encoder
  - Traing a variational autoencoder
- Probabilistic Topic Models
  - Latent Dirichlet Allocation
  - Estimating the LDA model

Research Example: Swedish Immigration Discourse

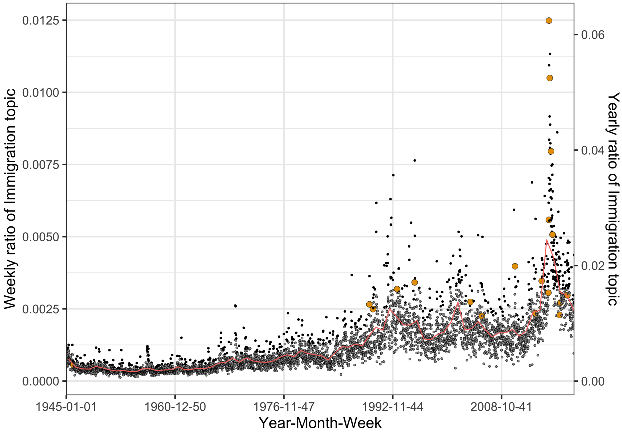


Figure: The Immigration topic in Swedish Newspapers (Hurtado Bodell et al, not in print)

Summary: Topic Models

- Topic models are unsupervised models for textual data

Summary: Topic Models

- Topic models are unsupervised models for textual data
- The Latent Dirichlet Allocation is a popular model

Summary: Topic Models

- Topic models are unsupervised models for textual data
- The Latent Dirichlet Allocation is a popular model
- A mixed membership model (a mixture of multinomial mixtures model)

Summary: Topic Models

UPPSALA UNIVERSITET

- Topic models are unsupervised models for textual data
- The Latent Dirichlet Allocation is a popular model
- A mixed membership model (a mixture of multinomial mixtures model)
- Use Gibbs samplers for estimation