Machine learning – Block 7

Måns Magnusson
Department of Statistics, Uppsala University

Autumn 2025

This week's lectures

- Variational autoencoders
- Probabilistic Topic Models
- (Diffusion models)
- Large Language Models

Section 2

Introduction

Why variational autoencoders and topic models?

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

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

UPPSALA
UNIVERSITET

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

- Used for:
  - Identify "closeness" in high-dimensional data

Use Cases

UPPSALA
UNIVERSITET

- 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
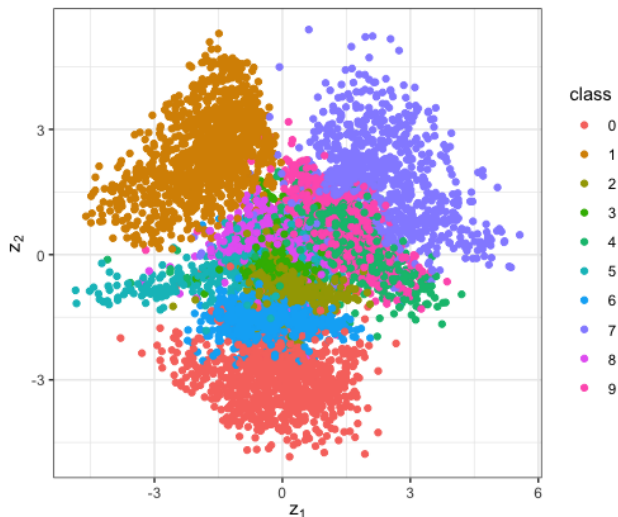
Figure: The latent state of MNIST using an Variational Autoencoder

Section 3

Autoencoders

Autoencoder

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

Autoencoder

- An autoencoder is a neural network (e.g. feed-forward) that takes an input x and predict (the same) x (r, reconstruction).
- Three parts:
  - encoder $f(x)$ (or $e(x)$)
  - code h
  - decoder $g(h)$ (or $d(h)$)
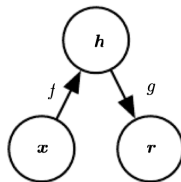
Figure: A Neural Autoencoder (Goodfellow et al, 2018)

Autoencoder loss (Reconstruction error)

For a deterministic autoencoder:

$$h = e_\phi(x), \qquad \hat{x} = d_\theta(h)$$

A common reconstruction loss is:

$$L(\theta, \phi) = \|x - \hat{x}\|_2^2 = \|x - d_\theta(e_\phi(x))\|_2^2$$

Minimizing squared error is equivalent (up to constants) to maximizing the log-likelihood of a Gaussian decoder with fixed variance:

$$x \mid h \sim \mathcal{N}(d_\theta(h), \sigma^2 I).$$

The Undercomplete Autoencoder

- More interesting: an undercomplete autoencoder:
  Dimension of code is lower than that of x
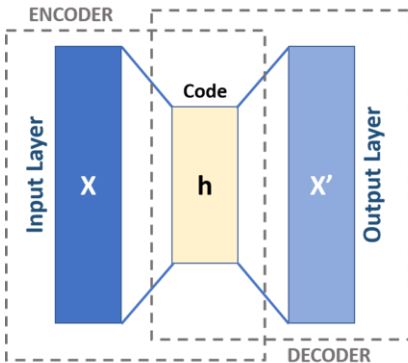
Figure: A Neural Autoencoder (Wikipedia)

PCA and autoencoders

UPPSALA
UNIVERSITET

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

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

$$L(\theta, \phi) = \sum_{i=1}^{N} \|x_i - W_\theta W_\phi x_i\|_2^2$$

PCA and autoencoders

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

$$L(\theta, \phi) = \sum_{i=1}^{N} \|x_i - W_\theta W_\phi x_i\|_2^2$$

- 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.

PCA and autoencoders

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

$$L(\theta, \phi) = \sum_{i=1}^{N} \|x_i - W_\theta W_\phi x_i\|_2^2$$

- 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 linear autoencoder (under squared loss and orthonormal constraints)
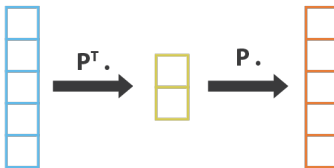
UPPSALA
UNIVERSITET

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

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 need to be regularized to not overfit the latent state

Probabilistic PCA as a decoder

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

Probabilistic PCA as a 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):

$$x_i \sim N(b + Wz_i, \sigma^2 I)$$

probabilistic PCA as an decoder

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

$$x_i \sim N(b + Wz_i, \sigma^2 I)$$

- Now, swap the simple parameters with a neural network

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

probabilistic PCA as an decoder

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

$$x_i \sim N(b + Wz_i, \sigma^2 I)$$

- Now, swap the simple parameters with a neural network

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

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

probabilistic PCA as an decoder

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

$$x_i \sim N(b + Wz_i, \sigma^2 I)$$

- Now, swap the simple parameters with a neural network

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

- This is an example of a Deep Latent Variable model (DLVM)
- One example of a DLVM 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

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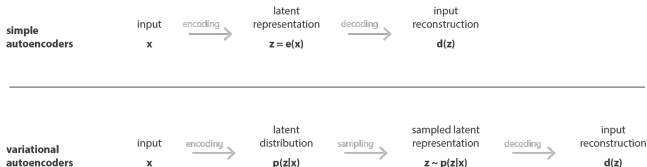
Figure: Autoencoder vs. the Variational Autoencoder (Rocca, 2019)

The Variational Autoencoder

Figure: The Variational Autoencoder (Kingma and Welling, 2018, Fig. 2.1)
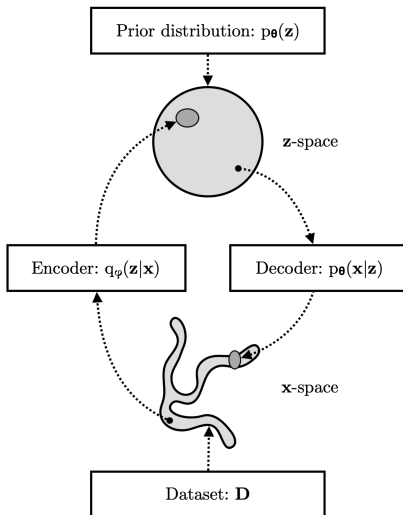
The Prior over the Latent Space

In a variational autoencoder, the latent variable is assumed to follow a simple prior:

$$z \sim p(z) = \mathcal{N}(0, I).$$

This prior defines where encoded data points are allowed to live in latent space.

- Prevents arbitrary or irregular latent encodings
- Encourages continuity and smooth interpolation
- Enables easy generation by sampling from $p(z)$

The probabilistic decoder

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

$$x_i \sim N(\text{NeuralNetwork}(z, \theta), \sigma_\theta^2 I)$$

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

The probabilistic decoder

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

$$x_i \sim N(\text{NeuralNetwork}(z, \theta), \sigma_\theta^2 I)$$

- x_i for observation i depends non-linearly on the latent state z_i
- A probabilistic linear decoder: pPCA
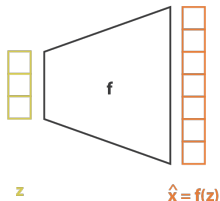
Figure: The Decoder (Rocca, 2019)

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:

$$z \sim N(\mu, \Sigma_\phi)$$

  where $\Sigma_\phi = \text{diag}\left(\sigma_{1,\phi}^2, \ldots, \sigma_{d,\phi}^2\right)$ (diagonal covariance) in practice.

The probabilistic encoder

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

- We assume that $q_\phi(z|x)$ follows a specific distribution. Commonly:

$$z \sim N(\mu, \Sigma_\phi)$$

where $\Sigma_\phi = \text{diag}\left(\sigma_{1,\phi}^2, \ldots, \sigma_{d,\phi}^2\right)$ (diagonal covariance) in practice.

- A neural network learns the parameters μ and Σ_ϕ

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

and

$$\log \sigma_{d,\phi}^2(x) = \text{NeuralNetwork}(x, \phi_\Sigma).$$

The probabilistic encoder

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

$$z \sim N(\mu, \Sigma_\phi)$$

  where $\Sigma_\phi = \text{diag}\left(\sigma_{1,\phi}^2, \ldots, \sigma_{d,\phi}^2\right)$ (diagonal covariance) in practice.

- A neural network learns the parameters μ and Σ_ϕ

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

  and

$$\log \sigma_{d,\phi}^2(x) = \text{NeuralNetwork}(x, \phi_\Sigma).$$

- Result: z_i for observation i depends non-linearly on x_i

Note: the decoder variance is typically fixed (e.g. $\sigma_\theta^2 I$), while the encoder learns a diagonal covariance.

The probabilistic encoder
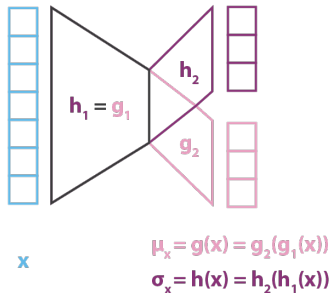
Figure: The Encoder (Rocca, 2019)

The Variational Autoencoder

Figure: The Variational Autoencoder (Rocca, 2019)

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

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
- Target:
- Maximize $\log p(x)$ (i.e. Explain the data as well as possible)

The Kullback–Leibler (KL) Divergence

What is it?

A measure of how well one probability distribution approximates another.

Definition

For two distributions p and q:

$$D_{KL}(p \parallel q) = \mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right]$$

Key properties

- $D_{KL}(p \parallel q) \geq 0$
- $D_{KL}(p \parallel q) = 0$ iff $p = q$
- Not symmetric: $D_{KL}(p \parallel q) \neq D_{KL}(q \parallel p)$

Interpretation

- Measures information lost when using q to approximate p
- Direction matters: *which distribution is the truth*

From the marginal likelihood to the ELBO

Goal: Compute or maximize the marginal log-likelihood

$$\log p_\theta(x)$$

Step 1: Introduce the latent variable

$$\log p_\theta(x) = \log \int p_\theta(x, z)\, dz$$

Step 2: Introduce a variational distribution

$$\log p_\theta(x) = \log \int q_\phi(z \mid x) \frac{p_\theta(x, z)}{q_\phi(z \mid x)}\, dz = \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right]$$

(This equality always holds as long as $q_\phi(z \mid x) > 0$.)

Step 3: Apply Jensen's inequality
(For a concave function f, $\quad f\big(\mathbb{E}[X]\big) \geq \mathbb{E}\big[f(X)\big]$)

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right]$$

The Evidence Lower Bound (ELBO)

Definition: The Evidence Lower Bound (ELBO) is

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right]$$

Key relationship:

$$\mathcal{L}_{\theta,\phi}(x) = \log p_\theta(x) - \mathrm{D}_{\mathrm{KL}}\big(q_\phi(z \mid x) \parallel p_\theta(z \mid x)\big)$$

(KL divergence is always non-negative.)

Interpretation:

- Maximizing the ELBO *increases* $\log p_\theta(x)$
- Minimizes the gap between $q_\phi(z \mid x)$ and $p_\theta(z \mid x)$
- The KL term regularizes the latent space by pushing $q_\phi(z \mid x)$ toward the prior $\mathcal{N}(0, I)$.

Important:

$$\mathcal{L}_{\theta,\phi}(x) \leq \log p_\theta(x), \quad \text{with equality iff } q_\phi(z \mid x) = p_\theta(z \mid x)$$

Optimizing the ELBO

Objective

We want to maximize the ELBO with respect to encoder and decoder parameters:

$$\max_{\theta,\phi} \mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x, z) - \log q_\phi(z \mid x) \right]$$

Optimizing the ELBO

Objective

We want to maximize the ELBO with respect to encoder and decoder parameters:

$$\max_{\theta,\phi} \mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x, z) - \log q_\phi(z \mid x) \right]$$

Problem

- The expectation is over $q_\phi(z \mid x)$
- We need gradients with respect to ϕ and θ
- Direct differentiation through sampling is not possible

Optimizing the ELBO

Objective

We want to maximize the ELBO with respect to encoder and decoder parameters:

$$\max_{\theta,\phi} \mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z \mid x)\right]$$

Problem

- The expectation is over $q_\phi(z \mid x)$
- We need gradients with respect to ϕ and θ
- Direct differentiation through sampling is not possible

Solution

- Approximate the expectation using Monte Carlo
- Rewrite the sampling step so gradients can flow (reparameterization trick)

The Reparameterization Trick

Assumption

The variational posterior is Gaussian:

$$q_\phi(z \mid x) = \mathcal{N}\big(\mu_\phi(x), \sigma_\phi^2(x)I\big)$$

Key idea

Instead of sampling $z \sim q_\phi(z \mid x)$, write:

$$z = g(\phi, x, \epsilon) = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Why this helps

- Randomness is now isolated in ϵ
- $z = g(\phi, x, \epsilon)$ is a differentiable function of ϕ
- Enables backpropagation through the sampling step

Monte Carlo Optimization of the ELBO

ELBO rewritten

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(x, g(\phi, x, \epsilon)) - \log q_\phi(g(\phi, x, \epsilon) \mid x)\right]$$

Monte Carlo approximation

Using one sample $\epsilon^{(1)} \sim \mathcal{N}(0, 1)$:

$$\mathcal{L}_{\theta,\phi}(x) \approx \log p_\theta(x, g(\phi, x, \epsilon^{(1)})) - \log q_\phi(g(\phi, x, \epsilon^{(1)}) \mid x)$$

Optimization

- Use stochastic gradient ascent (or descent on $-$ELBO)
- Gradients flow through $\mu_\phi(x)$ and $\sigma_\phi(x)$ and we can compute $\nabla \mathcal{L}_{\theta,\phi}(x)$
- One or few samples ϵ per datapoint is usually sufficient

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)

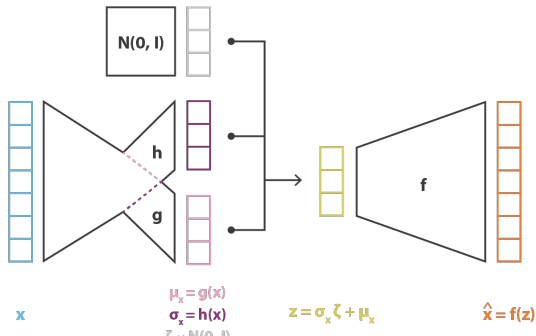# The Autoencoding Variational Bayes Algorithm

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

Summary

- Benefits of VAE:
    - Get a more interpretable latent state
    - We can estimate uncertainty (but its usually bad)
    - 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 (but its usually bad)
  - Regularize the latent state
  - We can inject knowledge in our latent state
- Still much ongoing research

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

VAE vs Diffusion Models

Variational Autoencoders (VAEs)

- Latent variable model: $z \sim p(z)$
- Trained by maximizing the ELBO
- Fast sampling (one forward pass)
- Compact, interpretable latent space
- Often produce blurrier samples

Diffusion Models

- No explicit low-dimensional latent space
- Learn to reverse a noise process
- Trained via denoising objectives
- Very high-quality samples
- Slow sampling (many steps)

Takeaway: VAEs prioritize efficient inference and representation learning, while diffusion models prioritize sample quality.

Section 5

Probabilistic Topic Models

UPPSALA
UNIVERSITET

• 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 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 distributional hypothesis
- Use cases:
  - Explore 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 distributional hypothesis
- Use cases:
  - Explore 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 distributional hypothesis
- Use cases:
  - Explore 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:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \text{ where } \mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

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

The Dirichlet Distribution

• Probability distribution over the simplex with K categories:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \text{ where } \mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

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

• 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:

$$f(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \text{ where } \mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)},$$

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

- 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 and

$$\mathbb{E}[x_i] = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j}$$

The Dirichlet Distribution
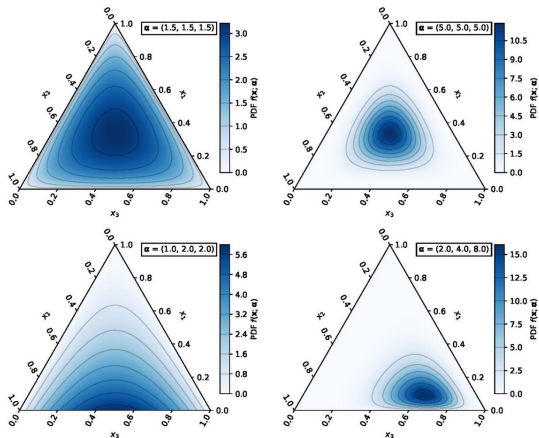
Figure: The Dirichlet Distribution (Wikipedia)

The distributional hypothesis

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

The distributional hypothesis

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

The distributional hypothesis

- Harris (1954) and Firth (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 Firth (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."
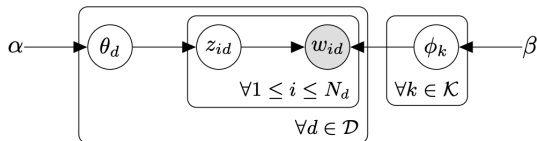2. "She is my friend indeed."

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 **z**, Θ and Φ

w$_1$	boat	shore	bank		
z$_1$	1	1	1		
w$_2$	Zlatan	boat	shore	money	bank
z$_2$	2	1	1	3	3
w$_3$	money	bank	soccer	money	
z$_3$	3	3	2	3	

Example of parameters **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 **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	

$$\Phi = \begin{array}{c|cccccc} & \text{boat} & \text{shore} & \text{soccer} & \text{Zlatan} & \text{bank} & \text{money} \\ \text{Topic 1} & 0.35 & 0.35 & 0.05 & 0.05 & 0.15 & 0.05 \\ \text{Topic 2} & 0.025 & 0.025 & 0.45 & 0.45 & 0.025 & 0.025 \\ \text{Topic 3} & 0.025 & 0.025 & 0.025 & 0.025 & 0.45 & 0.45 \end{array}$$

$$\Theta = \begin{array}{c|ccc} & \text{Topic 1} & \text{Topic 2} & \text{Topic 3} \\ \text{doc 1} & 0.96 & 0.02 & 0.02 \\ \text{doc 2} & 0.3 & 0.2 & 0.5 \\ \text{doc 3} & 0.05 & 0.35 & 0.6 \end{array}$$

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
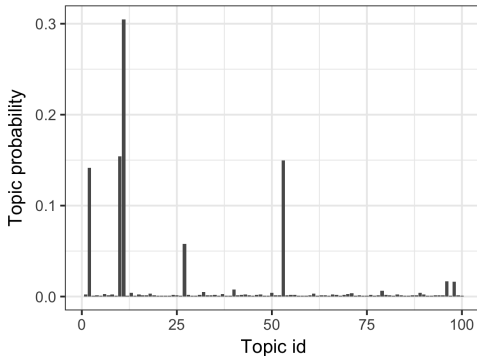leader of the Nassau County Democratic Party.

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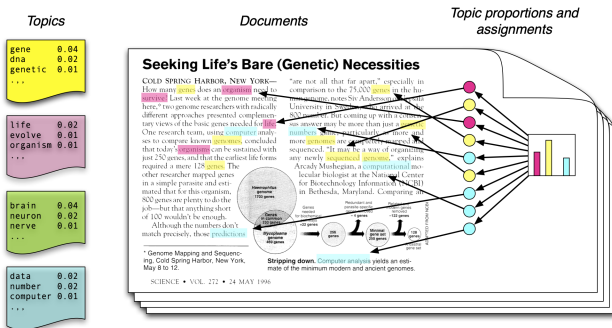
Figure: The Latent Dirichlet Allocation Model (Blei 2012, Fig. 1)

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

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

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

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	

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

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

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

Research Example: Swedish Immigration Discourse
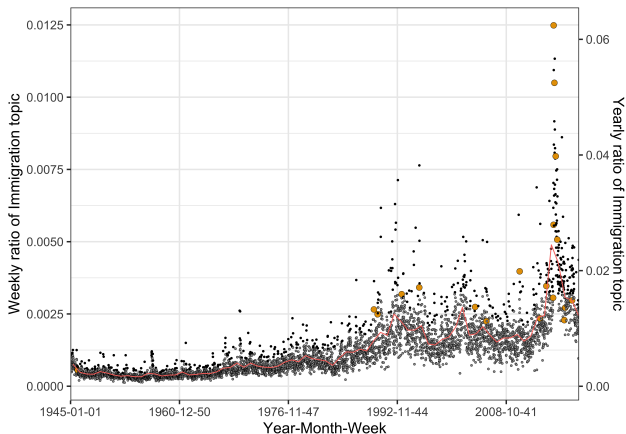
Figure: The Immigration topic in Swedish Newspapers (Hurtado Bodell et al, 2024)

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

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)

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)
- Use Gibbs samplers for estimation