Overview

1. Repetition

2. Kernel PCA

3. Autoencoding Variational Bayes
New Reference *

*Goodfellow, Bengio and Courville, Deep Learning
https://www.deeplearningbook.org/
Repetition
Factor Analysis Model

$z \in \mathbb{R}^k$ is a latent variable and $y$ is the observed data:

$$z \sim \mathcal{N}(0, 1)$$

$$x|z \sim \mathcal{N}(\mu + \Lambda z, \Psi)$$

Parameters of our model are thus:

- $\mu \in \mathbb{R}^n$
- $\Lambda \in \mathbb{R}^{n \times k}$
- Diagonal matrix $\Psi \in \mathbb{R}^{n \times n}$

**Note**: Dimensionality reduction since $k$ is chosen smaller than $n$. 
where $y$ are the observations.
Equivalent formulation

\[ z \sim \mathcal{N}(0, 1) \]
\[ \varepsilon \sim \mathcal{N}(0, \Psi) \]
\[ x = \mu + \Lambda z + \varepsilon \]

where \( \varepsilon \) and \( z \) are independent.

**Joint model:**

\[
\begin{bmatrix}
  z \\
  x
\end{bmatrix} \sim \mathcal{N}(\mu_{zx}, \Sigma)
\]

**Goal:** Identify \( \mu_{zx} \) and \( \Sigma \).
Factor Analysis

Joint model:

\[
\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} 1 & \Lambda^T \\ \Lambda & \Lambda \Lambda^T + \Psi \end{bmatrix} \right)
\]

Marginal Distribution

\[x \sim \mathcal{N} (\mu, \Lambda \Lambda^T + \Psi)\]

Log-Likelihood of parameters

\[l(\mu, \Lambda, \Psi) = \log \prod_{i=1}^{m} \frac{1}{(2\pi)^{\frac{n}{2}} |\Lambda \Lambda^T + \Psi|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x^{(i)} - \mu)^T (\Lambda \Lambda^T + \Psi)^{-1} (x^{(i)} - \mu) \right)\]
Motivation Represent face images efficiently and capture relevant information while removing nuisance factors like lighting conditions, facial expression, occlusion etc.

Idea

• Given training set of N images, use PCA to form a basis of K images, $K \ll N$.
• PCA for dimensionality reduction: Eigenface = eigenvector of covariance function
• Use lower dimensional features e.g. for face classification

Literature
Sirovich and Kirby, Low-dimensional procedure for the characterization of human face, 1987
Turk and Pentland, Face Recognition using Eigenfaces, CVPR 1991
Data *

Reconstruction*

Recall Exercise

Coding Exercise: Eigenfaces using CelebA dataset (> 200K celebrity images).

Sample Images

- Eyeglasses
- Bangs
- Pointy Nose
- Oval Face
- Wearing Hat
- Wavy Hair
- Mustache
- Smiling

Report and discuss (mean, speed, rotation, scaling, etc.) using piazza.
Solution to exercise

**Code**: [http://scikit-learn.sourceforge.net/0.8/auto_examples/applications/face_recognition.html](http://scikit-learn.sourceforge.net/0.8/auto_examples/applications/face_recognition.html)
**Problem Model Selection**

**Question**: How to choose the number of components?

- Number of components might be constrained by problem goal, computational or storage resources e.g. typically choose only 2 or 3 components for visualization problems.
- Eigenvalues magnitudes determine explained variance (recall Lec. 7). Search for elbow criterion.
Problem Non-linear Structure vs Random
Kernel PCA
Kernel PCA

*Wang, Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models, 2012
Solution using PCA

*Wang, Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models, 2012
Solution using Kernel PCA (polynomial) *

*Wang, Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models, 2012
Solution using Kernel PCA (Gaussian)*

*Wang, Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models, 2012
Robustness to noise

Original Data

Data with one random noise initialization

Linear PCA

Kernel PCA
Autoencoding Variational Bayes
Deep Generative Models

Implicit probabilistic models

• Do not specify the distribution of the data itself but rather a stochastic process which simulates the data
• Since they do not specify a distribution, they do not require a tractable likelihood
• One example are GAN’s (last week)

Explicit probabilistic models

• Specify model and use maximum likelihood
• Everything we have seen so far
• One example are Variational Autoencoders (today).
Motivation: Recent Examples using VAE Structures

Saito et.al, 3D Hair Synthesis Using Volumetric Variational Autoencoders, SIGGRAPH Asia 2018

https://www.youtube.com/watch?v=UT2EiLG4Mrg
\[ p(z \mid x, \alpha) = \frac{p(z, x \mid \alpha)}{\int p(z, x \mid \alpha)} \]  

(1)

Idea: Pick family of distributions over latent variables with its own variational parameter

\[ q(z \mid \nu) = \ldots? \]

and find variational parameters \( \nu \) such that \( q \) and \( p \) are "close".
Using Jensen’s inequality to obtain a lower bound

\[
\log p(x) = \log \int_Z p(z, x) = \\
= \log \int_Z p(z, x) \frac{q(z)}{q(Z)} \\
= \log \mathbb{E}_q \left( \frac{p(x, Z)}{q(Z)} \right) \\
\geq \mathbb{E}_q (\log p(x, Z)) - \mathbb{E}_q (\log q(Z))
\]

Proposal: Choose/design variational family \( Q \) such that the expectations are easily computable.
Relation with KL

\[
\text{KL}[q(z) \| p(z \mid y)] = \mathbb{E}_q \left[ \log \frac{q(Z)}{p(Z \mid y)} \right] \\
= \mathbb{E}_q[\log q(Z)] - \mathbb{E}_q[\log p(Z \mid y)] \\
= \mathbb{E}_q[\log q(Z)] - \mathbb{E}_q[\log p(Z, y)] + \log p(y) \\
= - (\mathbb{E}_q[\log p(Z, y)] - \mathbb{E}_q[\log q(Z)]) + \log p(y)
\]

Difference between KL and ELBO is precisely the log normalizer, which does not depend on \( q \) and is bounded by the ELBO.
Mean-field for conjugates

Mean-field: \( q(z, \beta) = q_{\lambda}(\beta) \prod_{i=1}^{k} q_{\varphi_i}(z_i) \)

- \(\lambda\) global variational parameter
- \(\varphi\) local variational parameter

Local update \( \varphi_i \leftarrow \mathbb{E}_\lambda[\eta_l(\beta, x_i)] \)

Global update: \( \lambda \leftarrow \mathbb{E}_\varphi[\eta_g(x, z)] \)

Note: Coordinate ascent iterates between local and global updates.
Summary

We use variational inference to approximate the posterior distribution

\[ \log p(x, \theta) = \text{ELBO}(q, \theta) + \text{KL}(q(z) \| p(z|x, \theta)), \]

\[ \log p(x, \theta) \geq \mathbb{E}_q[\log p(Z, x)] - \mathbb{E}_q[\log q(Z)] \]

To optimize the lower bound, we can use coordinate ascent!

**Problems:**

- In each iteration we go over all the data!
- Computing the gradient of the expectations above.

**Solution:** Stochastic and Black Box Variational Inference
Black-Box Stochastic Variational Inference in Five Lines of Python

David Duvenaud
dduvenaud@seas.harvard.edu
Harvard University

Ryan P. Adams
rpa@seas.harvard.edu
Harvard University

Abstract

Several large software engineering projects have been undertaken to support black-box inference methods. In contrast, we emphasize how easy it is to construct scalable and easy-to-use automatic inference methods using only automatic differentiation. We present a small function which computes stochastic gradients of the evidence lower bound for any differentiable posterior. As an example, we perform stochastic variational inference in a deep Bayesian neural network.
Gradient Estimates of the ELBO

\[
\text{ELBO} = \mathbb{E}_{q_{\nu}}[\log p_{\theta}(z, x)] - \mathbb{E}_{q}[\log q_{\nu}(z)]
\]

where \( \nu \) are the parameters of the variational distribution and \( \theta \) the parameters of the model (as before).

**Aim:** Maximize the ELBO

**Problem:** Need unbiased estimates of \( \nabla_{\nu, \theta} \text{ELBO} \).
Optimizing the ELBO

\[ \mathbb{E}_{q(\lambda)}[\log p(z, x) - \log q(z)] =: \mathbb{E}[g(z)] \]

**Exercise:**
\[ \nabla_\lambda \text{ELBO} = \nabla_\lambda \mathbb{E}[g(z)] = \mathbb{E}[g(z)\nabla \log q(z)] + \mathbb{E}[\nabla g(z)] \]

where \( \nabla \log q(z) \) is called the *score function*.

**Note:** The expectation of the score function is zero for any \( q \) i.e.

\[ \mathbb{E}_q[\nabla \log q(z)] = 0 \]

Thus, to compute a noisy gradient of the ELBO

- sample from \( q(z) \)
- evaluate \( \nabla \log q(z) \)
- evaluate \( \log p(x, z) \) and \( \log q(z) \)
Algorithm 1 Black Box Variational Inference

1: **Input:** data $x$, model $p(x,z)$, variational family $q_\varphi(z)$,
2: **while** Stopping criteria is not fulfilled **do**
3: Draw $L$ samples $z_l \sim q_\varphi(z)$
4: Update variational parameter using the collected samples

$$\varphi \leftarrow \varphi + \delta_t \frac{1}{L} \sum_{l=1}^{L} \nabla \log q(z_l)(\log p(x,z_l) - \log q(z_l))$$

5: Check step size and update if required!
6: **end while**

**Note:** Active research area (problem) is the reduction of the variance of the noisy gradient estimator.
Reparametrization trick

Simplified notation:

$$\nabla_{\nu} \mathbb{E}_{q_{\nu}}[f_{\nu}(z)]$$

Assume that there exists a fixed reparameterization such that

$$\mathbb{E}_{q_{\nu}}[f_{\nu}(z)] = \mathbb{E}_{q}[f_{\nu}(g_{\nu}(\varepsilon))]$$

where the expectation on the right does now not depend on $\nu$. Then

$$\nabla_{\nu} \mathbb{E}_{q}[f_{\nu}(g_{\nu}(\varepsilon))] = \mathbb{E}_{q}[\nabla_{\nu} f_{\nu}(g_{\nu}(\varepsilon))]$$

Solution: Obtain unbiased estimates by taking a Monte Carlo estimate of the expectation on the right.
Comparison

**Score Function** (reinforce)

- Differentiates the density $\nabla_\nu q(z, \nu)$
- Works for discrete and continuous models
- Works for large class of variational approximations
- Variance is a big issue

**Pathwise** (reparameterization)

- Differentiates the function $\nabla_z [\log p(x, z) - \log q(z, \nu)]$
- requires differentiable models
- requires variational models to have special form
- In practice better behaved variance

Variance Comparison

*NIPS Variational Inference Tutorial 2016
Visualization

GANs

GAN Progress

*Brundage et.al. 2018 https://img1.wsimg.com/blobby/go/3d82daa4-97fe-4096-9c6b-376b92c619de/downloads/1c6q2kc4v_50335.pdf
Progress GANs vs. State of the art VAE

**GAN**

![GAN Images](image1)

**Recent VAE**

![Recent VAE Images](image2)

Summary

Variational Autoencoders

- for image generation, necessity to reconstruct each pixel
- reparametrization is not applicable to discrete latent variables
- usually only allows to use a fixed standard normal as a prior
- images are often blurry compared to high-fidelity samples generated by GANs
- allows for efficient Bayesian inference

Generative Adversarial Networks

- Instability of training
- mode collapse i.e. generated samples are often only from a few modes of the data distribution
- only visual inspection since GANs do not support inference (can additionally train an inference network)
- does not support discrete visible variables

Generally: We are unable to control the attributes of generated samples e.g. aim for regularization which enforces disentangled latent codes.
Gan Zoo

Cumulative number of named GAN papers by month

*https://github.com/hindupuravinash/the-gan-zoo
Next week

So far missing

• Wake-Sleep Algorithm
• Independent Components
• Combination of State Space Models with Autoencoder
• Identification of number of latent components; Bayesian non-parametrics

Plan for Exercise on Tuesday:

Different Derivation based on Tutorial on Variational Autoencoders https://arxiv.org/abs/1606.05908
Questions?