Approximate Inference

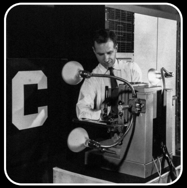
Creative Machine Learning - Course 07

Pr. Philippe Esling

esling@ircam.fr

Brief history of AI

1943 - Neuron
First model by McCulloch & Pitts (purely theoretical)



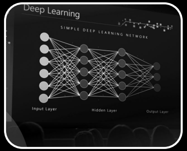
1957 - Perceptron
Actual **learning machine** built by Frank Rosenblatt
Learns character recognition analogically

1986 - Backpropagation
First to learn neural networks efficiently (*G. Hinton*)

1989 - *Convolutional NN*
Mimicking the vision system in cats (*Y. LeCun*)

This lesson

2012 - *Deep learning*
Layerwise training to have deeper architectures
Swoop all state-of-art in classification competitions



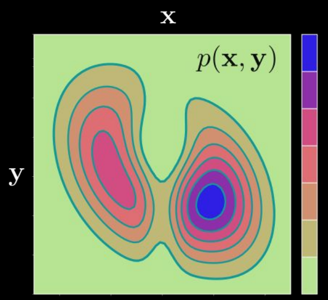
Lesson #6

2015 - *Generative model*
First wave of interest in generating data
Led to current model craze (VAEs, GANs, Diffusion)

2012 onwards Deep learning era

Pre-requisites for understanding generative models

1

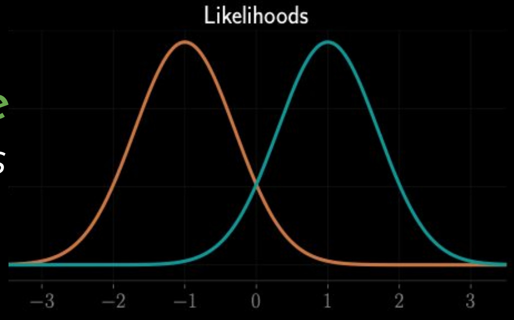


Probability theory

Random variables, distributions, independence

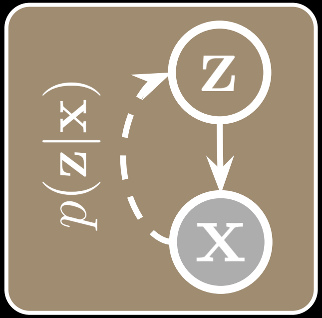
Bayesian inference

Bayes' theorem, likelihood, conjugate priors



2

3

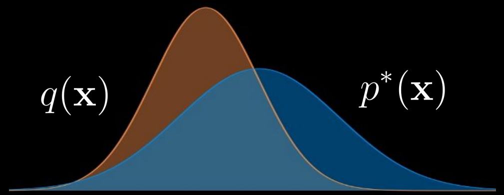


Latent models

Latent variables, probability graphs

Approximate inference

Latent variables, probability graphs



4

Lesson #6 2015 - *Generative model*

First wave of interest in generating data
Led to current model craze (VAEs, GANs, Diffusion)

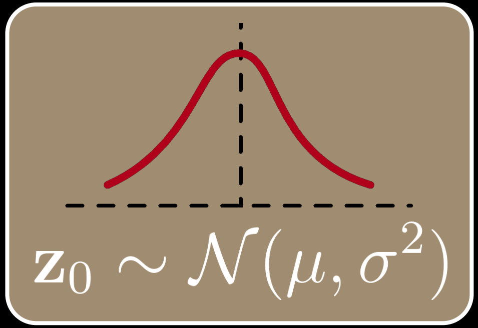
2012 onwards Deep learning era

Approximate inference

Probabilistic inference deals with simple distributions

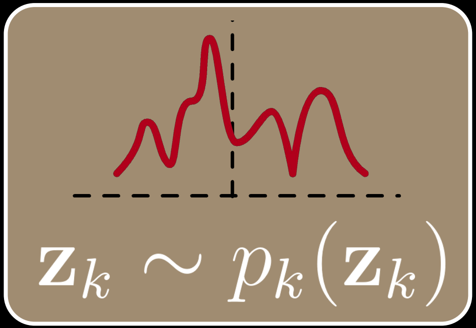
- Provide easier analytical solutions
- Implicit assumption of a simple explanation (capacity)

Issues
- Too simplistic assumption in most cases
- Real data follows largely complex distributions
- These distributions are usually intractable



The almighty Gaussian

Real data ?



How to model complex distributions but keep analytical simplicity ?

Entering the world of **approximate inference**

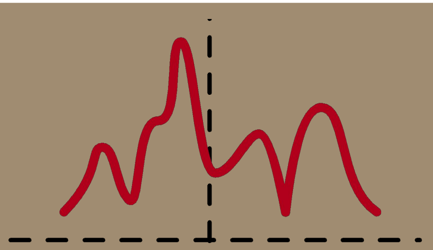
Approximating statistics
1. Sampling methods

Optimizing simpler distributions
1. Variational inference
2. Normalizing flows
3. Adversarial learning

Approximate inference

Question of the day

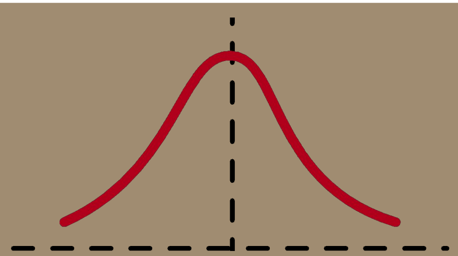
How can we model **real data** (of unknown distribution)
with only **simple tractable distributions**



$$\mathbf{z}_k \sim p_k(\mathbf{z}_k)$$

Real data ?

?



$$\mathbf{z}_0 \sim \mathcal{N}(\mu, \sigma^2)$$

The almighty Gaussian

1st approach: can we approximate some properties ?

Examples

- Estimate the expectation $\mathbb{E}_{\mathbf{z}_k}\left[p_k(\mathbf{z}_k)\right]$
- Optimize related functions $\underset{\mathbf{z}}{\operatorname{argmax}}\ f(\mathbf{z}_k)$

Sampling methods

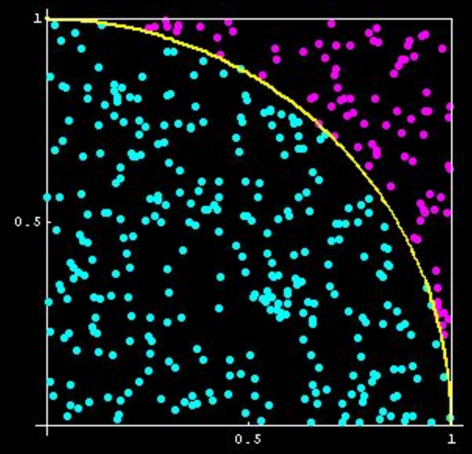
Family of *sampling methods*

- Considered the silver bullet of probabilistic modeling
- Very slow but *guaranteed to converge*
- Understand the limitations (when to use)

1 ***Monte-Carlo*** **methods**

- You have a complicated problem
- You cannot really work down the math
- Instead you can *simulate your problem* lots of time

Example Finding the value of π with a shotgun



$$\mathbb{E}\left[x^2 + y^2 \geq 1\right] \approx \frac{1}{M}\sum_{s=1}^{M}\left[x_s^2 + y_s^2 \geq 1\right]$$

with $x_s, y_s \sim \mathcal{U}(0,1)$

Monte-Carlo methods

Goal of Monte-Carlo methods | Estimate expected values by sampling

We define an *estimator* $\mathbb{E}_{p(\mathbf{x})}\left[f(\mathbf{x})\right] \approx \frac{1}{M}\sum_{s=1}^{M} f(\mathbf{x}_s)$

with $\mathbf{x}_s \sim p(\mathbf{x})$

Interesting properties

- This is an *unbiased* estimator
- Guarantees on the convergence

Then, we can **use this estimator** in more complex problems
For instance the **M-step** of the EM algorithm

$$\boxed{\max_{\theta} \mathbb{E}_q\left[\log p(\mathbf{x}, \mathbf{z}|\theta)\right]}$$

Monte-Carlo methods

Advantages

- Very simple to program
- Universally applicable to lots of problems
- Scalable to parallelization

Problems

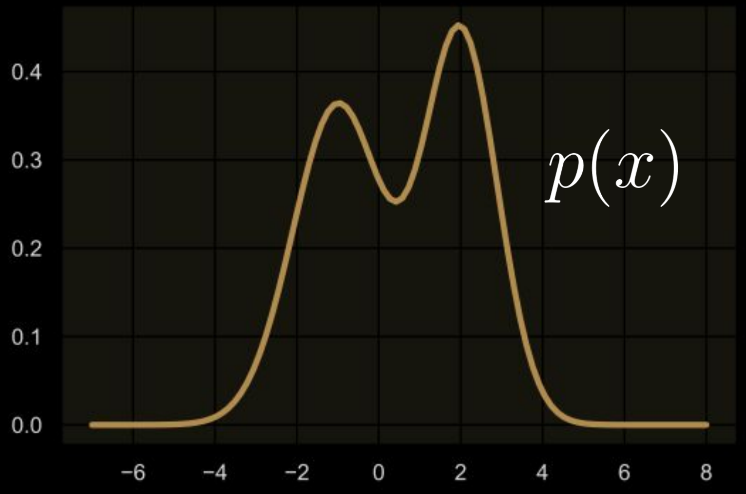
- Usually very slow (need lots of samples)
- Better solutions might exist

Major families of *sampling methods*

1. Rejection sampling
2. Markov Chain Monte-Carlo
    - Gibbs sampling
    - Metropolis-Hastings

Rejection sampling

Imagine that we have a *complex distribution* $p(x)$



We can upper-bound our distribution

Using approximation $q(x) = \mathcal{N}(1, 3)$

$$\boxed{p(x) \leq 2q(x)}$$

We use constant 2 to truly upper bound

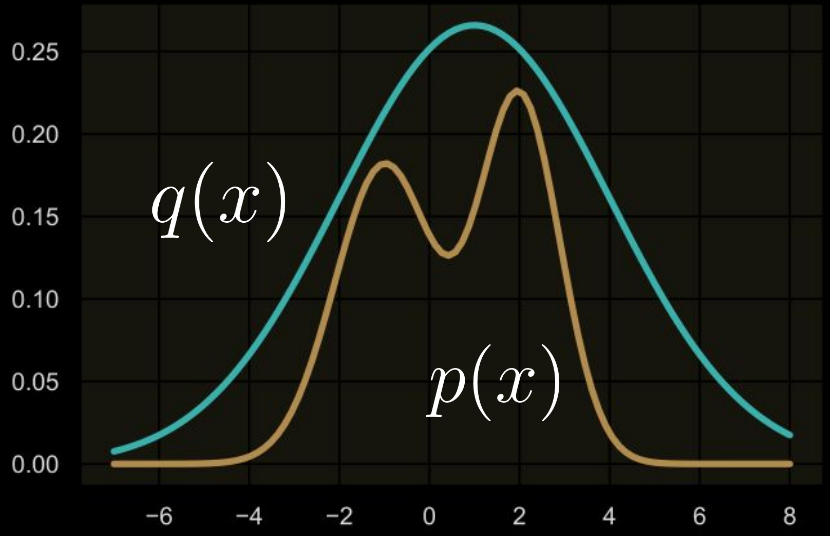
(upper bound distribution with multiplicative constant).

We can sample from $\hat{x} \sim q(x)$

How can we sample from $p(x)$?

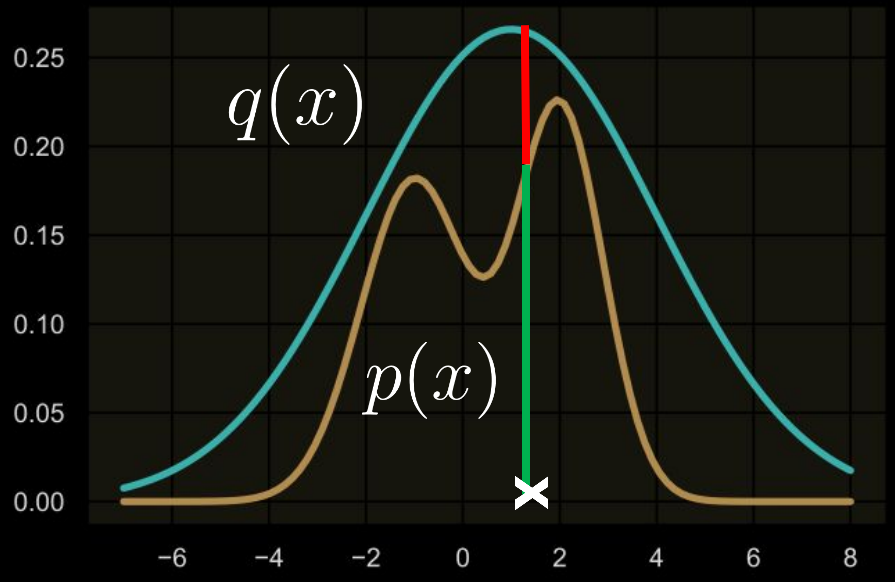
We know higher probability in $q(x)$

Need to ****reject**** some of the points



Rejection sampling

How do we reject some samples ?



We know it is proportional to the height

We can take a sample $\hat{x} \sim q(x)$

Take its coordinates $y \sim \mathcal{U}[0, 2q(\hat{x})]$

We accept $\hat{x}$ with probability

$$\boxed{\frac{p(x)}{2q(x)} \quad \text{if } \; y \geq p(x)}$$

We can even use this for unnormalized distributions as

$$\frac{\hat{p}(x)}{Z} \geq Mq(x) \longrightarrow \hat{p}(x) \geq ZMq(x)$$

Markov chains

Markov chain is a sequence of random variables $\{x_0, \cdots, x_n\}$
Such that x_i is independent of $x_0, \cdots, x_{i-2}$ given x_{i-1}

Markov property $\boxed{p(x_i|x_{i-1}, \cdots, x_0) = p(x_i|x_{i-1})}$

Typically described with $p(x_i|x_{i-1})$
Also noted as a *transition* $x_{k+1} \sim T(x_k \rightarrow x_{k+1})$

Stationary distribution

A distribution π is *stationnary* if it converges to a fixed distribution

$$\pi(x') = \sum_x \sim T(x \rightarrow x')\pi(x)$$

Gibbs sampling

Easiest method to *construct Markov chains to sample* from a distribution.

Base **unnormalized** distribution

$$p(x_1, x_2, x_3) = \frac{\hat{p}(x_1, x_2, x_3)}{Z}$$

Gibbs sampling

1. Start from a given **random** point (x_1^0, x_2^0, x_3^0)
2. Sample one dimension at a time
$$x_1^1 \sim p(x_1 | x_2 = x_2^0, x_3 = x_3^0) = \frac{\hat{p}(x_1, x_2^0, x_3^0)}{Z}$$
3. Re-apply this idea for next dimensions
$$x_2^1 \sim p(x_2 | x_1 = x_1^1, x_3 = x_3^0)$$
$$x_3^1 \sim p(x_3 | x_1 = x_1^1, x_2 = x_2^1)$$

Proof

$$p(x', y', z') = \sum_{x,y,z} q(x, y, z \rightarrow x', y', z') p(x, y, z)$$

Monte-Carlo methods

Advantages

- Very simple to program
- Universally applicable to lots of problems
- Scalable to parallelization

Problems

- Usually very slow (need lots of samples)
- Better solutions might exist

Sampling methods are extremely slow for complex problems

Can we turn this into an optimization problem ?

Entering the beautiful world of
variational inference

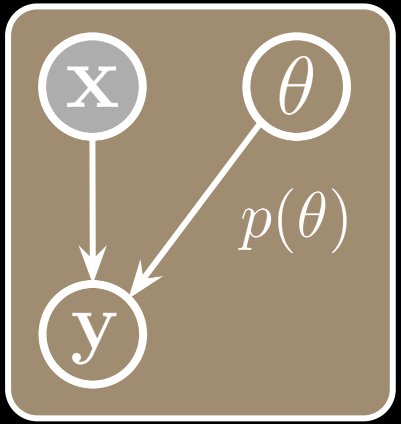
Variational inference

Goals

- Variational inference approximates distributions
- *Optimization trick* for parameter estimation
- Approximation technique for *intractable integrals*

Simple example - **Bayesian logistic regression**

Data



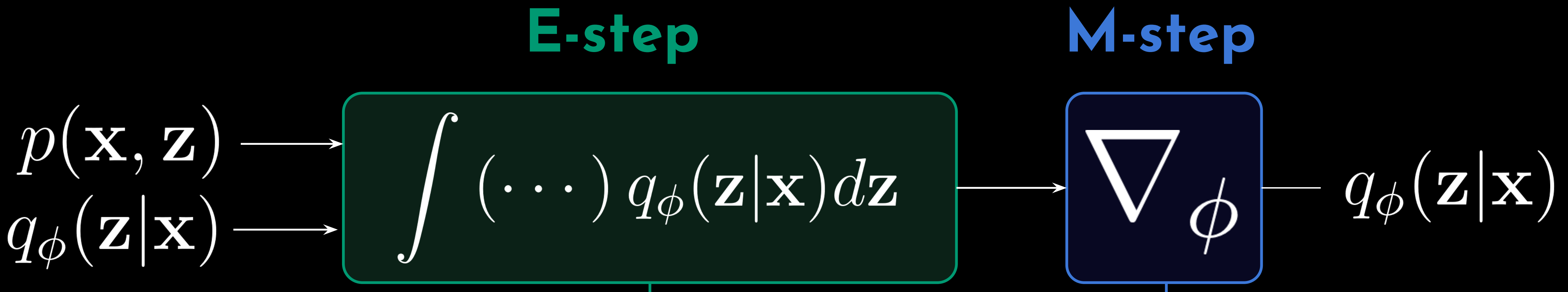
Parameters

Full distribution $$p(\mathbf{y}|\mathbf{x}) = \boxed{\int p(\mathbf{y}|\mathbf{x},\theta)d\theta}$$ **Intractable integral**

Classical inference approach

Recalling Expectation-Maximization

Can be summarized as the following operational flow



1. Compute expectations
2. Use to compute M-step gradients

Stochastic inference approach

Problems

- In general, *expectations are not known*
- Gradient is *of the parameters* of the distribution
- Expectation is taken on this distribution

$$\begin{matrix} p(\mathbf{x}, \mathbf{z}) \\ q_\phi(\mathbf{z}|\mathbf{x}) \end{matrix} \longrightarrow \boxed{\int (\cdots)\, q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z}} \longrightarrow \boxed{\nabla_\phi} \text{ — } q_\phi(\mathbf{z}|\mathbf{x})$$

1. Compute expectations
2. Use to compute M-step gradients

Stochastic inference approach

Problems

- In general, *expectations are not known*
- Gradient is *of the parameters* of the distribution
- Expectation is taken on this distribution

$$p(\mathbf{x}, \mathbf{z}) \longrightarrow$$
$$q_\phi(\mathbf{z}|\mathbf{x}) \longrightarrow$$

$$\nabla_\phi \longrightarrow \int (\cdots)\, q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} \;—\; q_\phi(\mathbf{z}|\mathbf{x})$$

Solution

- Use an *approximate* distribution
- Perform *optimization* on its parameters
- Then we can compute expectation

Stochastic gradient estimator

All approaches can be summarized as

$$\nabla_{\phi}\mathbb{E}_{q_{\phi}(\mathbf{z})}\left[f_{\theta}(\mathbf{z})\right] = \nabla \int \boxed{q_{\phi}(\mathbf{z})} \boxed{f_{\theta}(\mathbf{z})} d\mathbf{z}$$

Two potential solutions

Score-function estimator
Differentiate the density $q_{\phi}(\mathbf{z})$

Pathwise gradient estimator
Differentiate the function $f_{\theta}(\mathbf{z})$

Score-function estimator

If we focus on differentiating the **density**

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[f_\theta(\mathbf{z})\right] = \nabla \int \boxed{q_\phi(\mathbf{z})} f_\theta(\mathbf{z}) dz$$

$$= \mathbb{E}_{q_\phi(\mathbf{z})}\left[f_\theta(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z})\right]$$

Gradient reweighted by the value of the function

Other names

- Likelihood-ratio trick
- REINFORCE algorithm
- Automated inference
- Black-box inference

When to use

- Function is not differentiable
- Density is easy to sample from
- Density is known and differentiable

Pathwise gradient estimator

If we focus on differentiating the **function**

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[f_\theta(\mathbf{z})\right] = \nabla \int q_\phi(\mathbf{z}) \boxed{f_\theta(\mathbf{z})} d\mathbf{z}$$

$$= \mathbb{E}_{p(\epsilon)}\left[\nabla_\phi f(g_\phi(\mathbf{x}, \epsilon))\right]$$

Introducing *reparameterization*

Find an invertible function that expresses the input as a transformation of a base distribution $\mathbf{z} = g_\phi(\epsilon) \quad \epsilon \sim p(\epsilon)$

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[f_\theta(\mathbf{z})\right] = \mathbb{E}_{p(\epsilon)}\left[f(g_\phi(\mathbf{x}, \epsilon))\right]$$

Pathwise gradient estimator

If we focus on differentiating the **function**

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[f_\theta(\mathbf{z})\right] = \nabla \int q_\phi(\mathbf{z}) \boxed{f_\theta(\mathbf{z})} d\mathbf{z}$$

$$= \mathbb{E}_{p(\epsilon)}\left[\nabla_\phi f(g_\phi(\mathbf{x}, \epsilon))\right]$$

Other names

- Reparameterisation trick
- Stochastic backprop
- Perturbation analysis

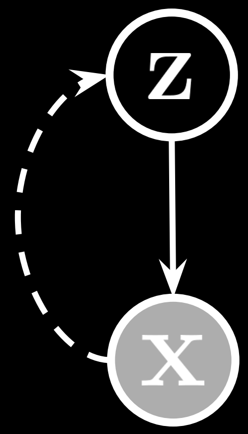
When to use

- Function is differentiable
- Density can be approximated
- Easy to sample from base distrib

Variational inference

We take back our inference problem $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$

posterior ----- $p(\mathbf{z}|\mathbf{x})$



$p(\mathbf{z})$ -------- **prior**

$p(\mathbf{x}|\mathbf{z})$ ----- **likelihood**

$$\boxed{p(\mathbf{x}) = \int_{\mathbb{R}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}}$$

How can we transform this intractable integral

Optimize a simpler and tractable distribution $q(\mathbf{z})$?

Variational inference

Starting from intractable integral $$p(\mathbf{x}) = \int_{\mathbb{R}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$$

Goal of Variational Inference

Approximate this by using **optimization** instead of **derivation**.

Select a family of *parametric* distributions $q_\phi \in \mathcal{Q}$

Optimize the parameters ϕ in order to solve

$$\underset{\phi}{\operatorname{argmin}}\ \mathcal{D}_{KL}\left[q_\phi \| p\right]$$

Modeling the joint probability

Deriving the variational objective

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$$

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}q(\mathbf{z}|\mathbf{x})dz$$

Introducing $q_\phi \in \mathcal{Q}$

$$\geq \int q(\mathbf{z}|\mathbf{x}) \log \left(p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right) dz$$

Jensen's inequality
$f\left(\mathbb{E}_{q_\phi(\mathbf{z})}\left[\mathbf{x}\right]\right) \geq \mathbb{E}_{q_\phi(\mathbf{z})}\left[f\left(\mathbf{x}\right)\right]$

$$\geq \underbrace{\int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z})dz}_{\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]} - \underbrace{\int q(\mathbf{z}|\mathbf{x}) \log \left(\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right) dz}_{\mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right]}$$

$$\boxed{\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{reconstruction}} - \underbrace{\mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right]}_{\text{regularization}}}$$

Variational inference

Optimizing our final objective

$$\boxed{\mathcal{L}(\boldsymbol{\theta}, \phi)} = \boxed{\underbrace{\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{reconstruction}}} - \boxed{\underbrace{\mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right]}_{\text{regularization}}}$$

Optimizing reconstruction

Monte-Carlo estimator $\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \frac{1}{L}\sum_{l=1}^{L} \log p(\mathbf{x}^{(l)}|\mathbf{z})$

... too much variance and hard to train

Reparametrization trick $\mathbf{z} = g_\phi(\boxed{\mathbf{x}}, \boxed{\epsilon})$

deterministic part stochastic part

$$\boxed{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[f_\theta(\mathbf{z})\right] = \mathbb{E}_{p(\epsilon)}\left[f(g_\phi(\mathbf{x}, \epsilon))\right] = \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\mathbf{x}, \epsilon^{(l)}))}$$

Kullback-Leibler divergence

Optimizing regularization with KL divergence

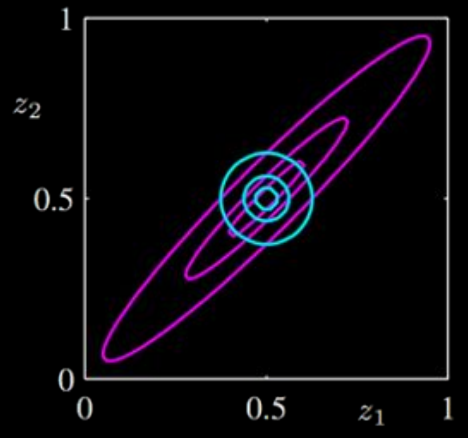
$$\mathcal{D}_{KL}\left[p(\mathbf{z}) \parallel q(\mathbf{z})\right] = \int p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z}$$

Properties of KL

$$\mathcal{D}_{KL}\left[p(\mathbf{z}) \parallel q(\mathbf{z})\right] \geq 0. \ \forall p, q$$

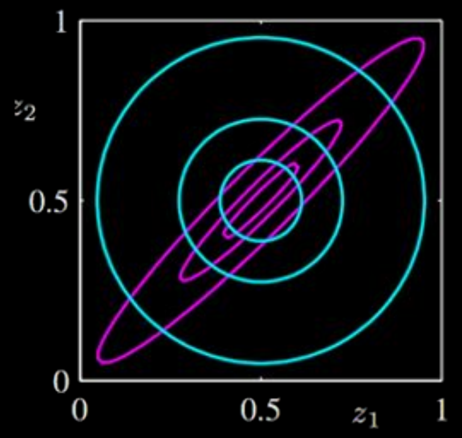
$$\mathcal{D}_{KL}\left[p(\mathbf{z}) \parallel q(\mathbf{z})\right] \neq \mathcal{D}_{KL}\left[q(\mathbf{z}) \parallel z(\mathbf{z})\right]$$

$$\mathcal{D}_{KL}\left[p(\mathbf{z}) \parallel q(\mathbf{z})\right] = 0 \text{ only if } p = q$$



Assymetric (divergence)

$$\mathcal{D}_{KL}\left[p(\mathbf{z}) \parallel q(\mathbf{z})\right] \neq \mathcal{D}_{KL}\left[q(\mathbf{z}) \parallel z(\mathbf{z})\right]$$



Mean-field approximation

How to choose $q_\phi \in \mathcal{Q}$

- flexible
- easy to sample from
- simple closed form

Mean-field family $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(\mathbf{z}_j)$

- Each dimension is an independent distribution
  - Usually a Normal distribution
- Each dimension has its own factor of variation
- Compatible with conjugacy

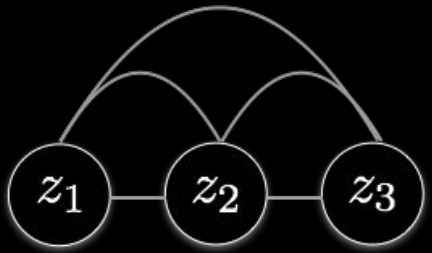
$$p(\mathbf{z}) = p(z_1)p(z_2)p(z_3)$$

Approximation families

Least expressive

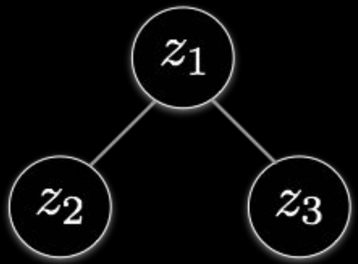
Mean-field family

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(\mathbf{z}_j)$$



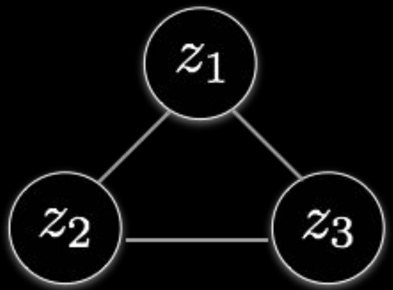
Auto-regressive

$$q(\mathbf{z}) = \prod_{i=1}^{K} q(z_i | z_1, \cdots, z_{i-1})$$



Structured

$$q(\mathbf{z}) = q(z_1)q(z_2|z_1)q(z_3|z_1)$$



True posterior

Most expressive