Deep Learning Seminar

Chapter 19 Approximate Inference

Heechul Lim

Department of Information and Communication Engineering

DGIST

2017.01.31

Contents

- **Chapter 19 Approximate Inference**

19.1 Inference as Optimization

19.2 Expectation Maximization

19.3 MAP Inference and Sparse Coding

19.4 Variational Inference and Learning

19.5 Learned Approximate Inference

Variational inference in the book

$$\log p_\theta(x^{(1)}, \cdots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$$

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

$$\log p_\theta(x^{(i)}) \geq \mathbb{E}_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p_\theta(x, z)\right] = \mathcal{L}(\theta, \phi; x^{(i)})$$

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} \mathcal{L}(\theta, \phi; x^{(i)}).$$

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Machine learning and Algorithm

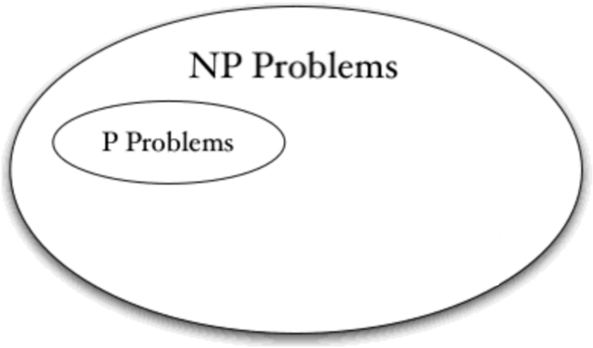
- **Understanding of algorithms**
  - Good algorithms and bad algorithms
  - Problems that can be solved or cannot be solved
- **Understanding of these things**
  - Big O notation
  - P and NP
  - Reduction
  - NP Complete problem
  - Approximation algorithm

Big O notation

- The algorithm should have three things
  - Input data
  - Output
  - Purpose
- We need a good algorithm, not an arbitrary algorithm
- How to distinguish between good or bad
  - Big O notation. e.g. $O(n)$
  - n is size of input
  - $O()$ is upper limit time

P and NP

- Polynomial time. e.g. $O(n^2)$
- Exponential time. e.g. $O(e^n)$
- P problem
  - A problem that can present an algorithm that can solve the problem in polynomial time
- NP problem
  - A problem that can be used to distinguish whether a given solution is a polynomial time solution or not
- P is the subset of NP



https://ko.wikipedia.org/wiki/P-NP_문제

Approximation

- Reduction from Problem X to Y
  - If we have an algorithm that solves Problem Y,
    we can find an algorithm that can solve problem X using this
- NP-Complete(NPC) problem
  - All NP problems can be reduced to the problem
  - When solving the problem of machine learning,
    there are many cases where the problem is an NPC problem
- Needs for ways to solve NPC problem
- The approximation algorithm
  - It does not provide an exact answer,
    but it does get an approximated solution in the polynomial
  - When the original algorithm gave the answer x,
    the α $-$approximation algorithm gives the answer αx

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Decision rule

$$p(C|X) = \frac{p(X|C)p(C)}{p(X)}$$

x: Input data
C: Classes of data

http://sanghyukchun.github.io/61/

Frequentists V.S. Bayesians

- **Frequentists**
  - Probability is given as a **frequency**
  - The **parameters** are unknown, but **fixed constants**
  - Good estimates can be obtained through **many trials**
- **Bayesians**
  - Probability is **degree of faith**
  - The **parameters** are **random variables**
  - A good estimates should be obtained **only with given data**

Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)

- **MLE**

$$\hat{\theta} = \arg\max_{\theta}[Pr(\mathbf{x}_{i=1,\cdots,I})] = \arg\max_{\theta}[\prod_{i=1}^{I} Pr(\mathbf{x}_i|\theta)]$$

- **MAP**

$$\hat{\theta} = \arg\max_{\theta}[\prod_{i=1}^{I} Pr(\mathbf{x}_i|\theta)Pr(\theta)]$$

- **Bayesian approach**

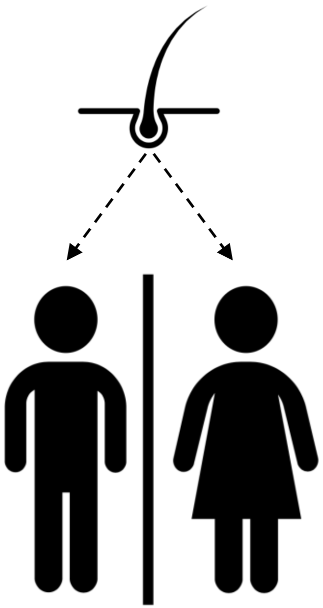
$$Pr(\theta|\mathbf{x}_{1,\cdots,I}) = \frac{\prod_{i=1}^{I} Pr(\mathbf{x}_i|\theta)Pr(\theta)}{Pr(\mathbf{x}_{1,\cdots,I})}$$

x: Input data
θ: Parameters

http://www.navisphere.net/2541/computer-vision-models-learning-and-inference-4-fitting-probability-models/

Simple example about MLE and MAP

- **Classification**
  - Look at the x (length) of the hair falling on the floor, and classify whether the hair is from c (male or female)

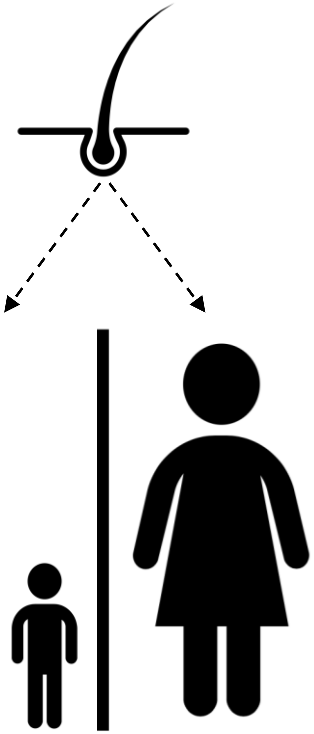


MLE method
A method of choosing a high probability (sex) by comparing the probability that such a hair will come from a man, $\boldsymbol{p\,(x\,|\,male)}$, and the probability that such hair will come from a woman, $\boldsymbol{p\,(x\,|\,female)}$

MAP method
A method of choosing a high probability (sex) by comparing the probability of finding a male x, $\boldsymbol{p\,(male\,|\,x)}$, and female probability $\boldsymbol{p\,(female\,|\,x)}$

Simple example about MLE and MAP

- **Solve this problem in an area where the sex ratio is uneven**
- **MAP can find a more accurate model than MLE**



MLE method

$\boldsymbol{p(x \mid male), p(x \mid female)}$

$$p(x|female) = \frac{p(x, female)}{p(female)}$$

MAP method

$\boldsymbol{p(male \mid x), p(female \mid x)}$

$$p(female|x) = \frac{p(female, x)}{p(x)} = \frac{p(female, x)}{p(x, female) + p(x, male)}$$

$$= \frac{p(female, x)}{p(x|female)p(female) + p(x|male)p(male)}$$

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Inference

- **Inference stage**
  - Learn the model to compute $\mathrm{p}(C|X)$ using training data
- **Decision stage**
  - Make the actual class assignment decision
    using the posterior probability computed at the inference stage
- **Discriminant function**
  - Directly map the decision to input x without above process

Generative V.S. Discriminative

- **Generative**
  - To model the joint probability and to make a decision using the result of 'generate' the samples into the distribution
  - Preliminary assumption is needed
  - E.g. Gaussian Mixture Model, Restricted Boltzmann Machine
- **Discriminative**
  - To directly compute the posterior class probability $p(C|X)$ in the inference stage
  - E.g. Logistic regression, SVM, Boosting, Neural networks

$$\boxed{p(C|X)} = \frac{\boxed{p(X|C)p(C)}}{p(X)}$$

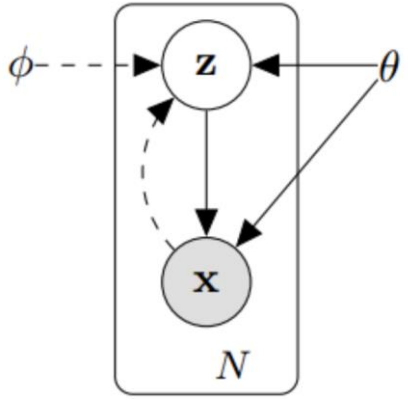
http://sanghyukchun.github.io/61/

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Problem scenario

- **Directed graph**
- **Problem**
  - **We need to know posterior $p_\theta(z|x)$ but $p_\theta(x)$ is intractable**
  - **Because we can't marginalize the $p_\theta(x)$ for every z**



$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$

x: Input data
θ, ϕ: Parameters
Arrow: $p_\theta(x|z)p_\theta(z)$

z: Latent variables
Dotted arrow: $p_\theta(z|x)$ approxiated using $q_\phi(z|x)$

http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-1.html

Taxonomy of deep generative models

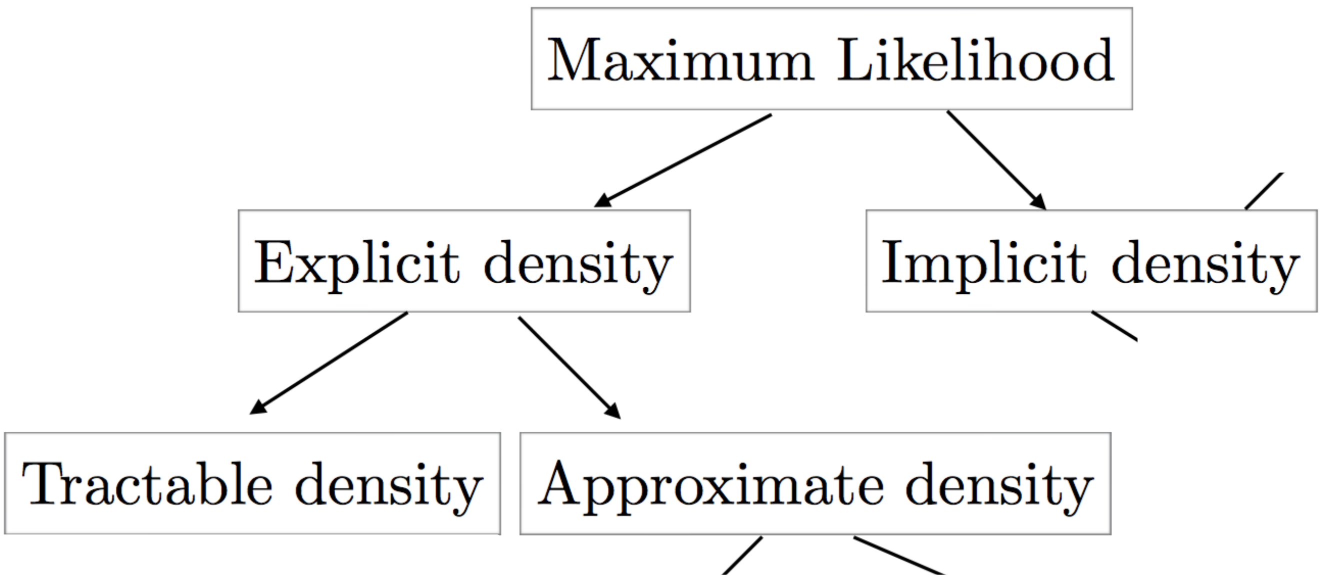
$$\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$

Maximum Likelihood

https://arxiv.org/abs/1701.00160

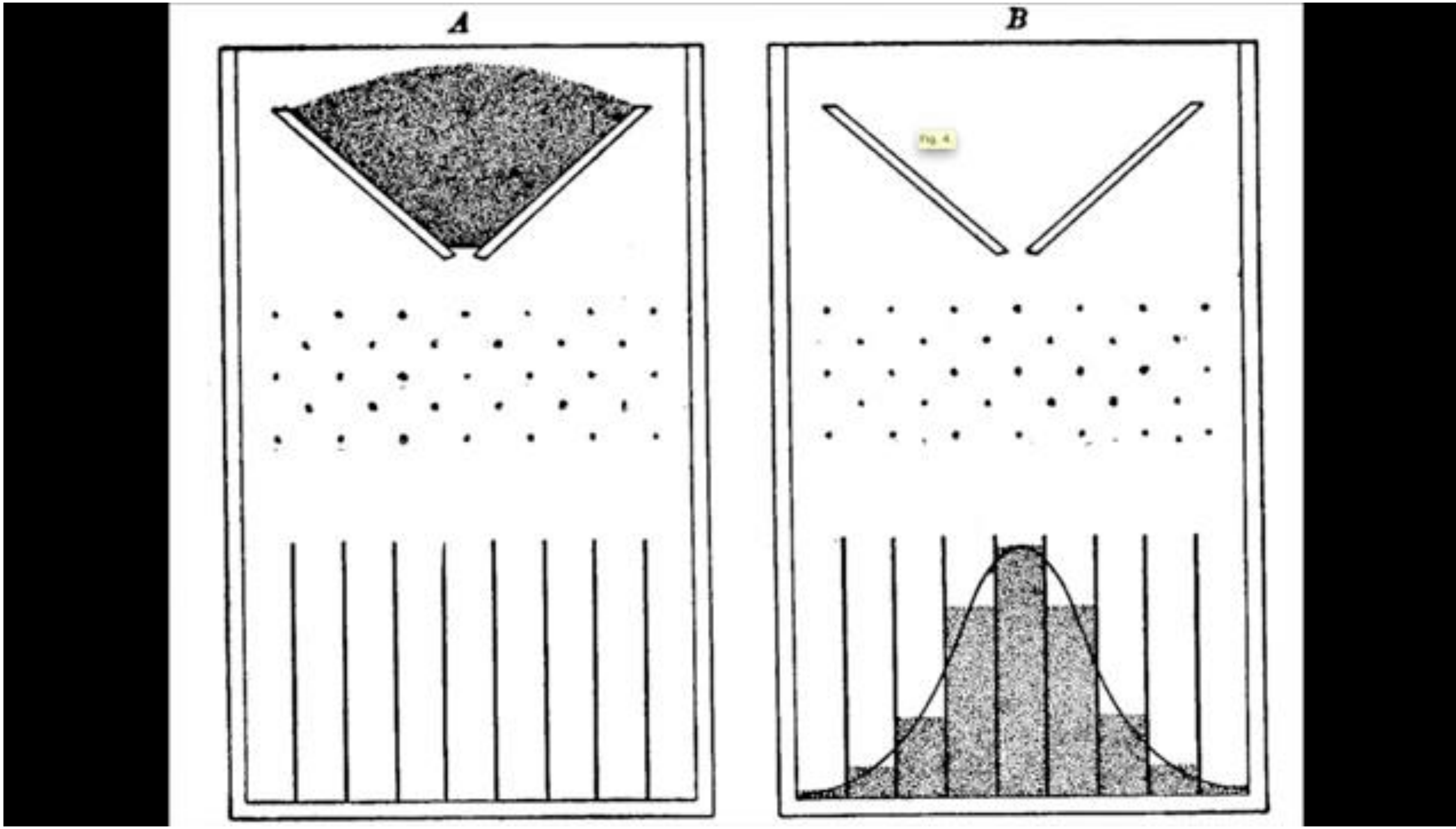
Taxonomy of deep generative models

$$\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$



https://arxiv.org/abs/1701.00160

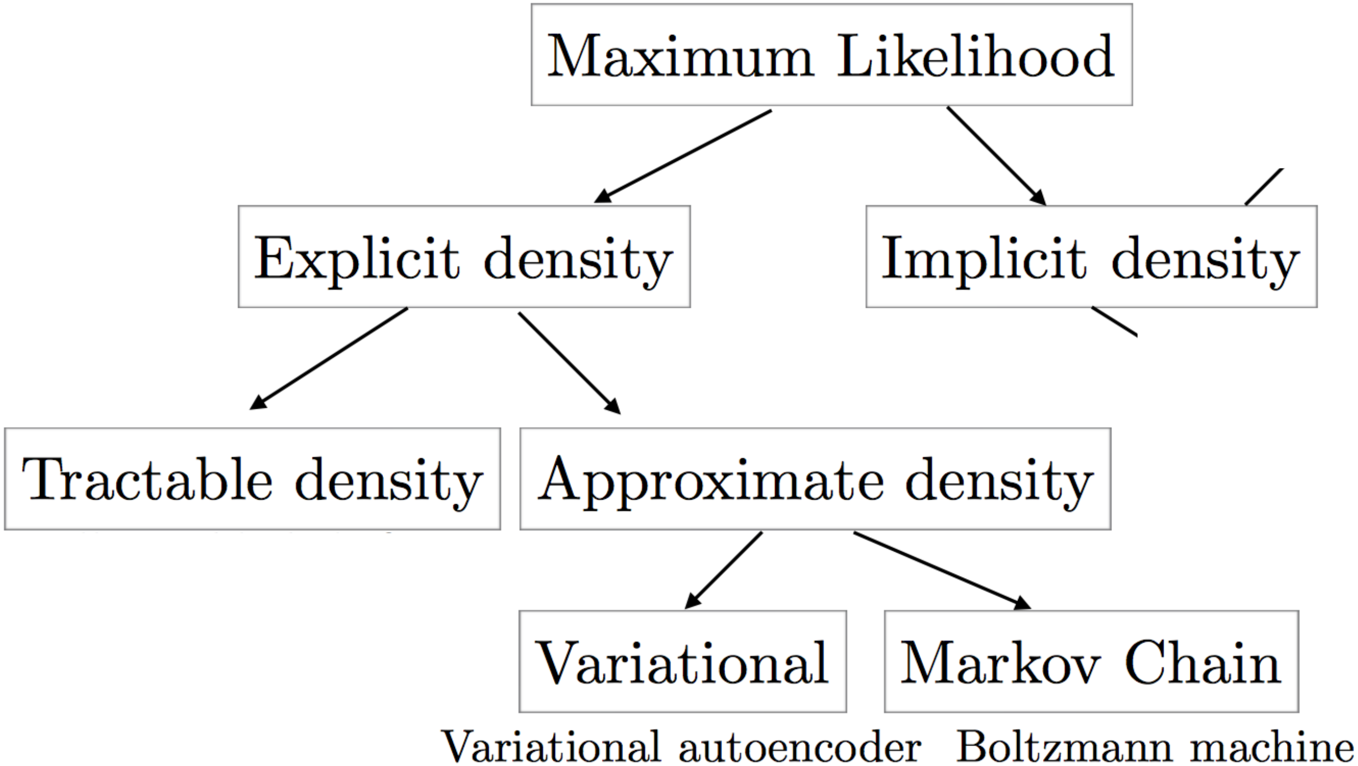
Taxonomy of deep generative models



https://youtu.be/Ws63I3F7Moc?t=2m44s

Taxonomy of deep generative models

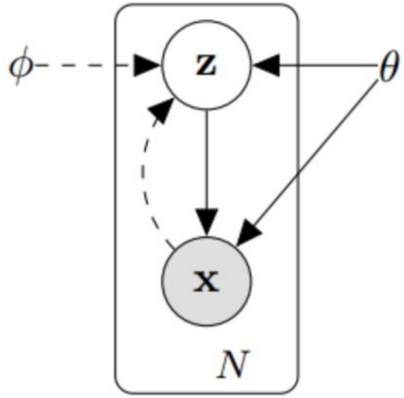
$$\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$



https://arxiv.org/abs/1701.00160

Variational approximation

- **It changes the posterior to an easier matter**
  - Evidence Lower Bound(ELBO)



$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$

x: Input data
θ, ϕ: Parameters
Arrow: $p_\theta(x|z)p_\theta(z)$

z: Latent variables
Dotted arrow: $p_\theta(z|x)$ approxiated using $q_\phi(z|x)$
$q_\phi(z|x)$: alternative model of $p_\theta(z|x)$

$$\mathcal{L}(x;\theta) \leq log_{p_{model}}(x;\theta)$$

http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-1.html

Calculus of variations

- **Variational method is a field of calculus**
- **Unlike general calculus, it deals with functional**
- **It deals with derivative, which deals with the functions that maximize or minimize any value**

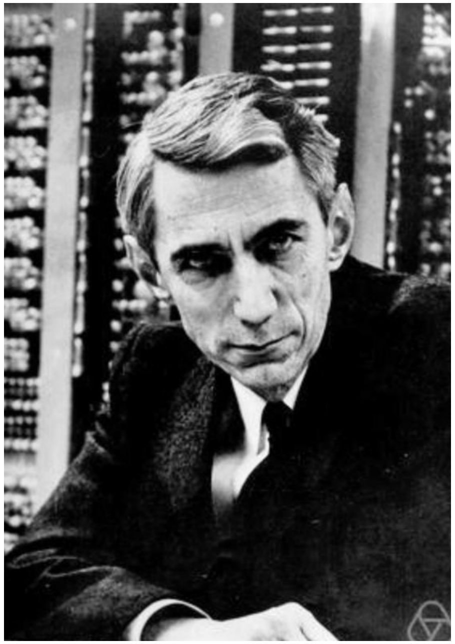
$$E.g. \quad KL(Q_\phi(Z|X)||P(Z|X))$$

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

KL-Divergence

- **Information theory**
  - **It is introduced by Claude Elwood Shannon**
  - **It quantifies the value of information**
  - **It is defined using a probability function**



$$h(x) = -\log\{p(x)\}$$

The value of information of **the first prize** at lotto

$$h(x) = -\log_2\left(\frac{1}{8{,}145{,}060}\right) \cong 23$$

The value of information of **the fifth class** at lotto

$$h(x) = -\log_2\left(\frac{1}{45}\right) \cong 5.5$$

https://en.wikipedia.org/wiki/Claude_Shannon
https://www.slideshare.net/lovelykihohong/variational-inference-intro-korean-ver

KL-Divergence

- **Entropy**
  - **The average amount of information a system has**
  - **It is possible to compare system V.S. system**

$$H(x) = -\sum_{x} p(x) log p(x)$$

https://www.slideshare.net/lovelykihohong/variational-inference-intro-korean-ver

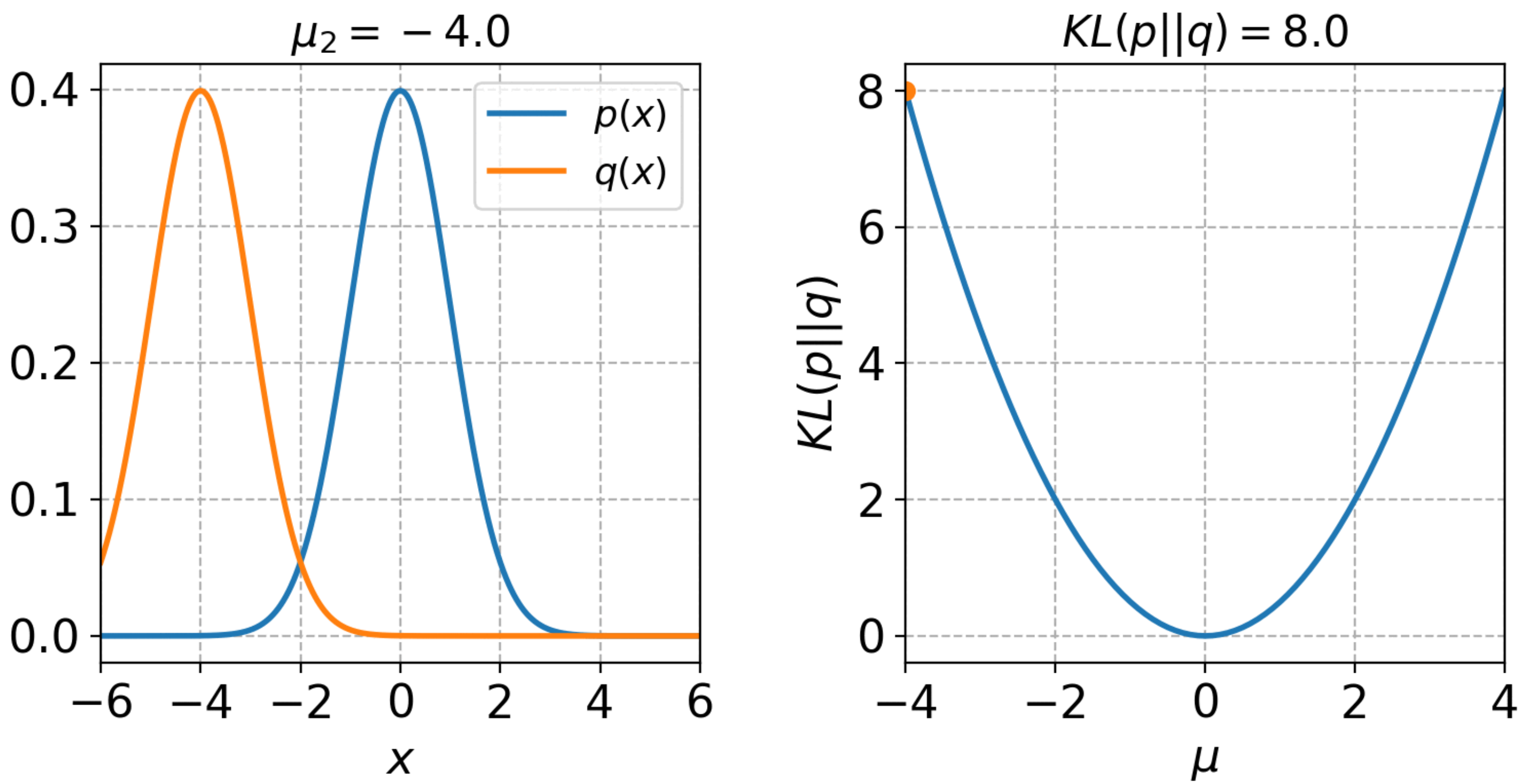
KL-Divergence

- **Definition**
  - There is data generated from P probability distribution
  - Assuming that this is from the Q probability distribution, this is the amount of additional information that is generated
- **Scenario**
  - Samples $\leftarrow p(x)$
  - $p(x)$ is intractable, so we need to introduce $q(x)$ instead of $p(x)$
  - The difference between $p(x)$ and $q(x)$ is KLD value

$$KL(p\|q) = -\int p(x)\ln q(x) - \left(-\int p(x)\ln p(x)dx\right)$$
$$= -p(x)\ln\left[\frac{q(x)}{p(x)}\right]dx$$

http://sanghyukchun.github.io/62/

KL-Divergence



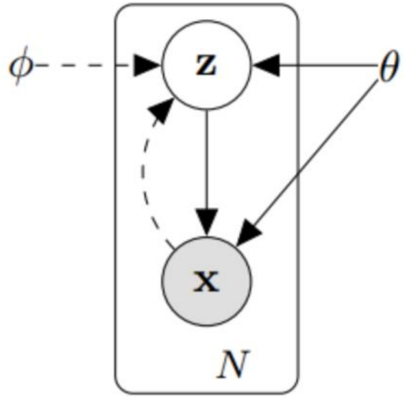
https://qiita.com/ceptree/items/9a473b5163d5655420e8

Contents

- Approximation
- Maximum Likelihood(MLE) and Maximum A Posteriori(MAP)
- Inference
- Taxonomy of deep generative models
- KL-Divergence
- Variational Inference

Variational approximation

- **It changes the posterior to an easier matter**
  - Evidence Lower Bound(ELBO)



$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)dz}$$

x: Input data
θ, ϕ: Parameters
Arrow: $p_\theta(x|z)p_\theta(z)$

z: Latent variables
Dotted arrow: $p_\theta(z|x)$ approxiated using $q_\phi(z|x)$
$q_\phi(z|x)$: alternative model of $p_\theta(z|x)$

$$\mathcal{L}(x;\theta) \leq log_{p_{model}}(x;\theta)$$

http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-1.html

Variational inference

- Make an inference with the **model Q_ϕ** that we know **as close as** possible to P_θ
- Use **KL-Divergence** to calculate the difference

$$KL(Q_\phi(Z|X)||P(Z|X)) = \sum_{z \in Z} q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)}$$

http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-2.html

Optimization problem

- **KL-Divergence decomposition**

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

https://www.slideshare.net/HyungjooCho2/deep-generative-modelpdf

Optimization problem

- **Transformation**
  - A problem that was a **statistical inference** that estimates posterior into an **optimization problem**
- **Optimization problem**
  - To solve the **MLE**, solving the **optimization problem** by adding a **regularization term that minimizes the difference** between p_θ and q_ϕ

$$\log p_\theta(x^{(1)}, \cdots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$$

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

$$\log p_\theta(x^{(i)}) \geq \mathbb{E}_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p_\theta(x, z)\right] = \mathcal{L}(\theta, \phi; x^{(i)})$$

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} \mathcal{L}(\theta, \phi; x^{(i)}),$$

http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-2.html

EM Algorithm

X: Observed data tensor
Z: Hidden data tensor
$p()$: Probability distribution
$q()$: Probability distribution of Z
θ: Parameter

- **Maximum likelyhood estimation**

$$\max_{\theta} p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, Z|\theta)$$

- **Maximum log likelyhood estimation**

$$\ln p(\mathbf{X}|\theta) = L(q, \theta) + KL(q||p)$$

$$L(q, \theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} \right\}$$

$$KL(q||p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})} \right\}$$

http://norman3.github.io/prml/docs/chapter09/4 http://sanghyukchun.github.io/70/

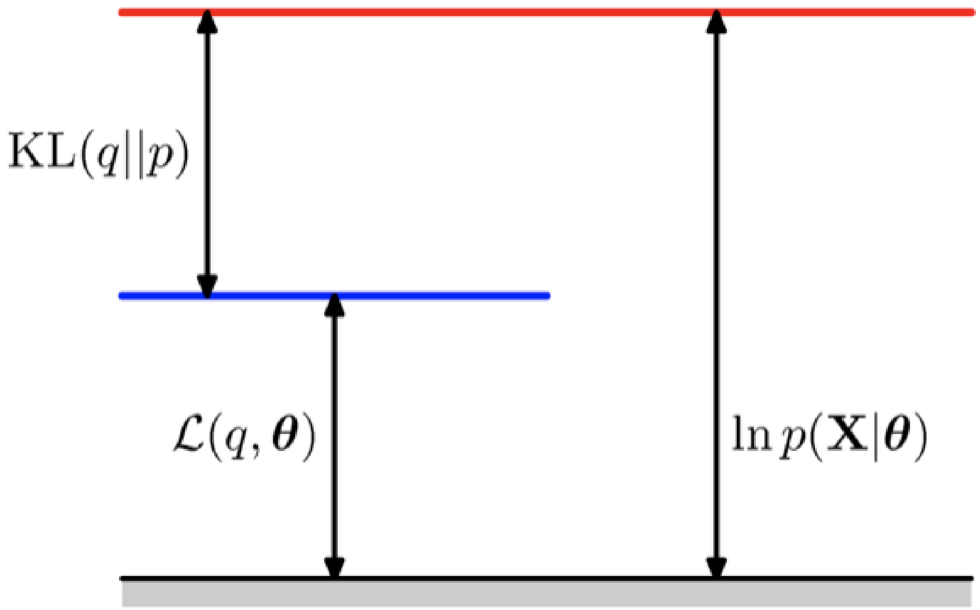
EM Algorithm

X: Observed data tensor
Z: Hidden data tensor
$p()$: Probability distribution
$q()$: Probability distribution of Z
θ: Parameter

- **Maximum likelyhood estimation**

$$\ln p(\mathbf{X}|\theta) = L(q, \theta) + KL(q||p)$$

- **Current status**



http://norman3.github.io/prml/docs/chapter09/4 http://sanghyukchun.github.io/70/

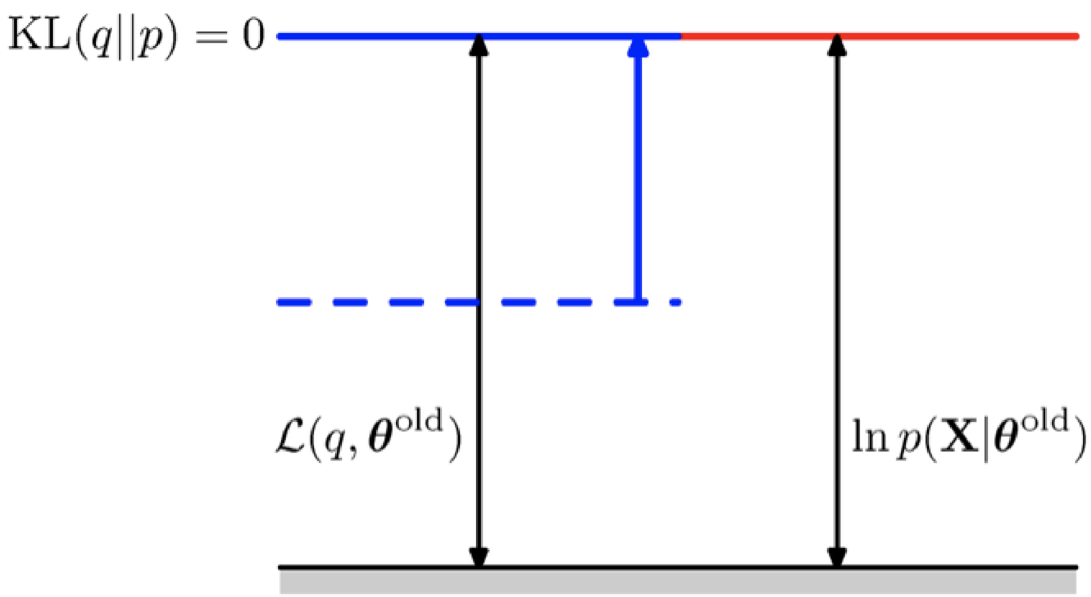
EM Algorithm

X: Observed data tensor
Z: Hidden data tensor
$p()$: Probability distribution
$q()$: Probability distribution of Z
θ: Parameter

- **Maximum likelyhood estimation**

$$\ln p(\mathbf{X}|\theta) = L(q,\theta) + KL(q||p)$$

- **E-step**



http://norman3.github.io/prml/docs/chapter09/4 http://sanghyukchun.github.io/70/

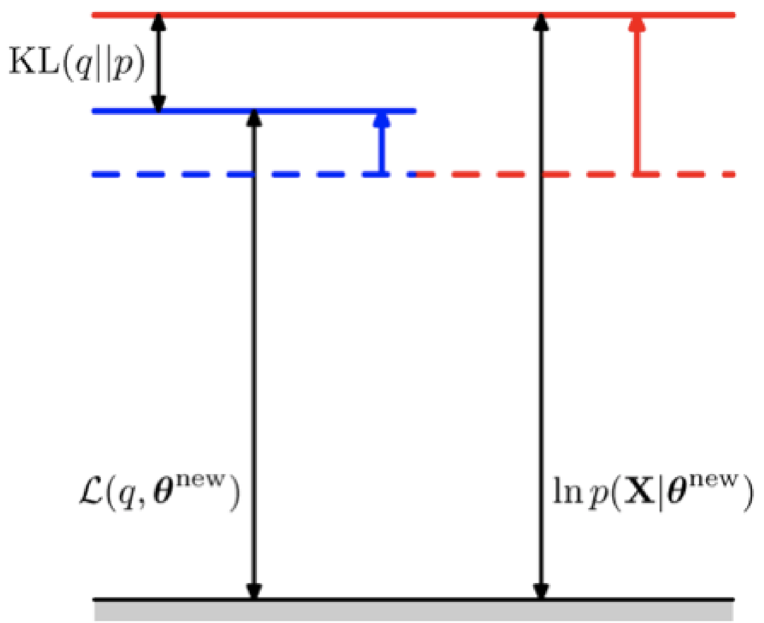
EM Algorithm

X: Observed data tensor
Z: Hidden data tensor
$p()$: Probability distribution
$q()$: Probability distribution of Z
θ: Parameter

- **Maximum likelyhood estimation**

$$\ln p(\mathbf{X}|\theta) = L(q, \theta) + KL(q||p)$$

- **M-step**

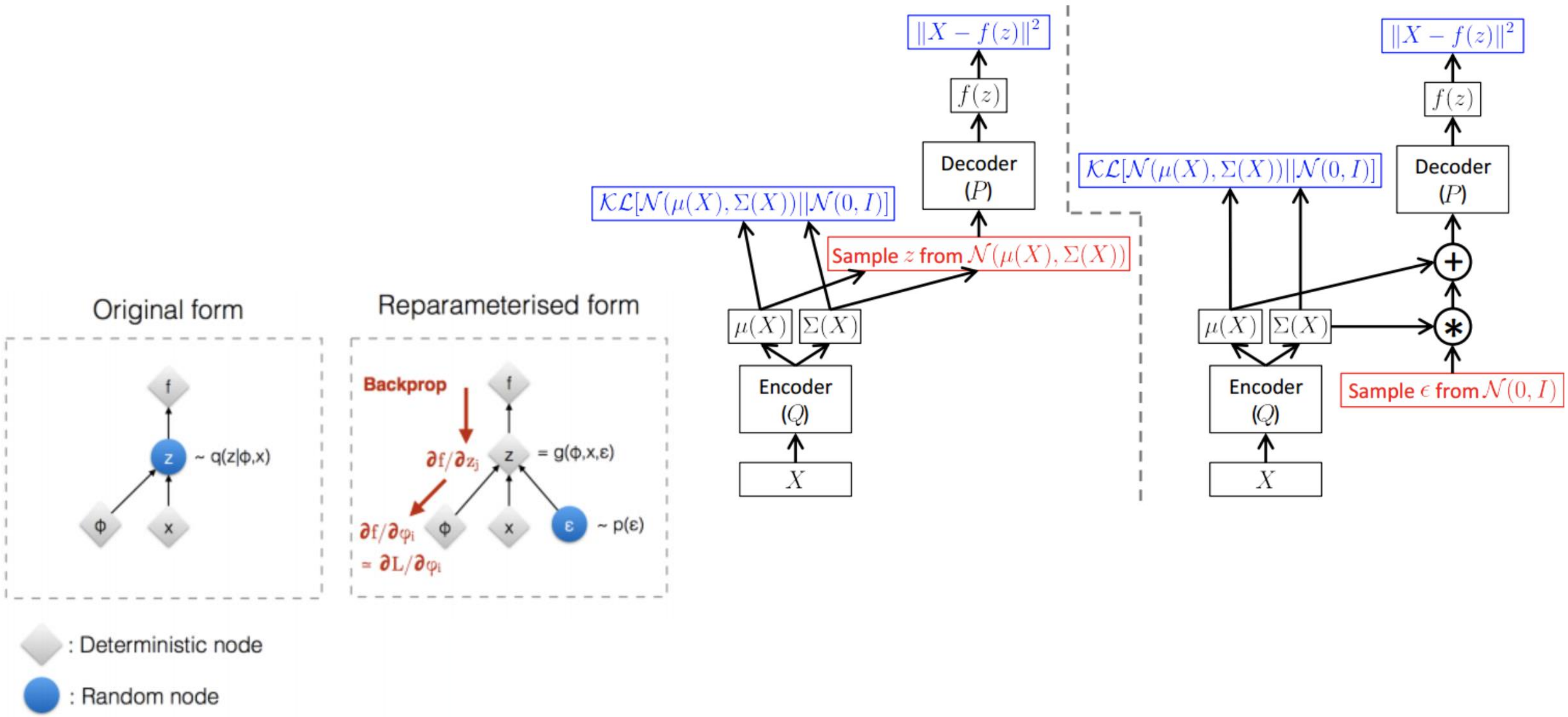


http://norman3.github.io/prml/docs/chapter09/4 http://sanghyukchun.github.io/70/

Reparametric trick for backpropagation

$$\log p_\theta(x^{(i)}) \geq \mathbb{E}_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p_\theta(x, z)\right] = \mathcal{L}(\theta, \phi; x^{(i)})$$

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} \mathcal{L}(\theta, \phi; x^{(i)}),$$

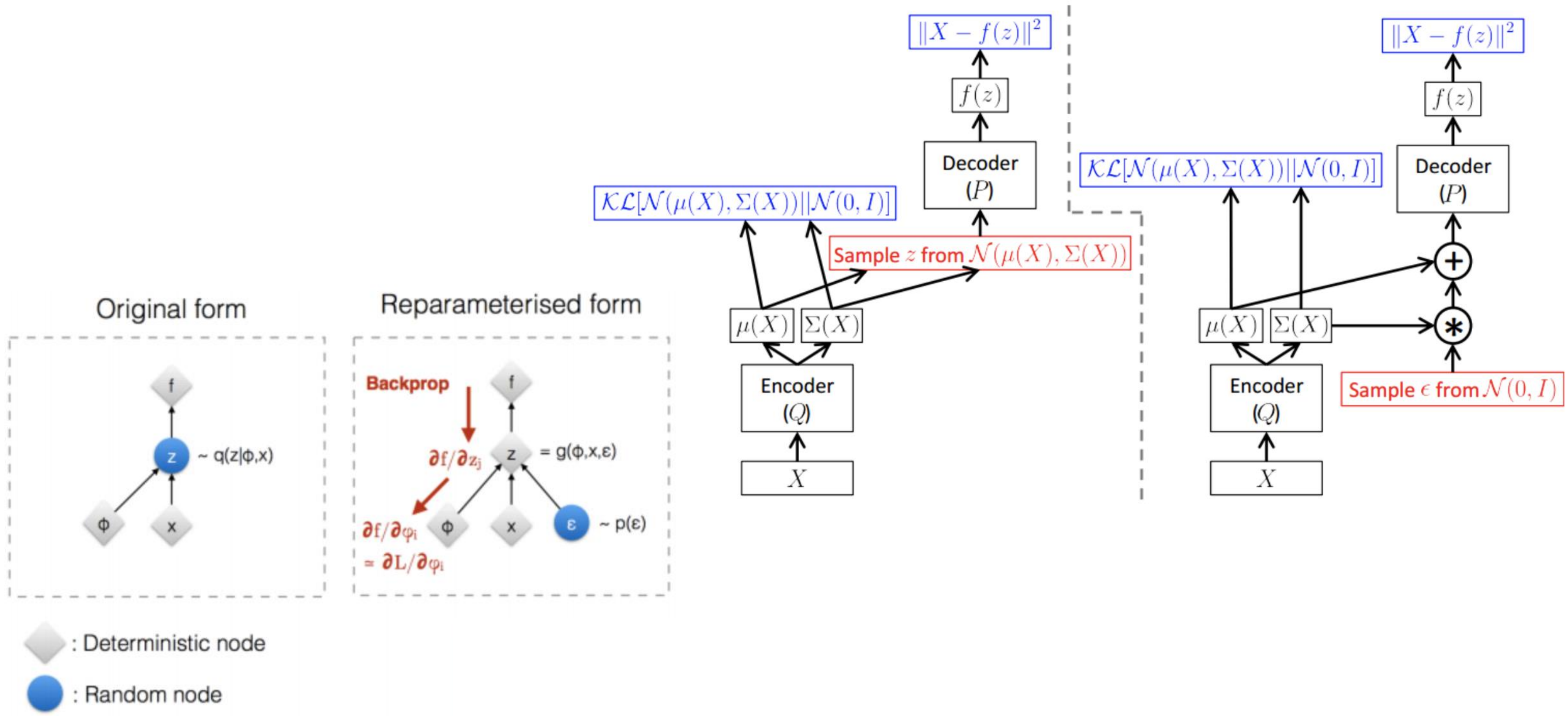


http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-3.html

Reparametric trick for backpropagation

$$\log p_\theta(x^{(i)}) \geq \mathbb{E}_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p_\theta(x,z)\right] = \mathcal{L}(\theta,\phi;x^{(i)})$$

$$(\theta^*,\phi^*) = \arg\max_{\theta,\phi} \mathcal{L}(\theta,\phi;x^{(i)}),$$

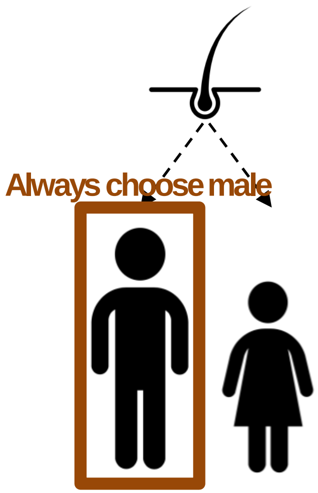


http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-3.html

Thank you

Decision rule

$$\frac{p(C)}{p(X)}$$



- For example, if a man is 60% and a woman is 40%
- Always choosing the male class is best choice
- But this is not good result,
  because the error rate is always 0.4

- If we have more information $P(X|C)$ about this,
  we can make more accurate model

$$C = argmaxP(C|X) = argmax\frac{P(X|C)P(C)}{\sum_i P(X|C_i)P(C_i)}$$

$$P(error|x) = \begin{cases} P(C1|X) \text{ if we decide C2} \\ P(C2|X) \text{ if we decide C1} \end{cases}$$