Speech and Language Processing

Lecture 4

Variational inference and sampling

Information and Communications Engineering Course

Takahiro Shinozaki

2020/10/20

Lecture Plan (Shinozaki's part)

I gives the first 6 lectures about speech recognition. Through these lectures, the backbone of the latest speech recognition techniques is explained.

1. 10/19 (remote)
   Speech recognition based on GMM, HMM, and N-gram
2. 10/19 (remote)
   Maximum likelihood estimation and EM algorithm
3. 10/20 (remote)
   Bayesian network and Bayesian inference
4. 10/20 (remote)
   Variational inference and sampling
5. 10/22 (remote)
   Neural network based acoustic and language models
6. 10/22 (remote)
   Weighted finite state transducer (WFST) and speech decoding

Variational inference

Variational Bayes

Evaluating posterior distribution is often not feasible

➡ Let's approximate it with a simpler distribution

$$p(\Lambda \mid D) = \frac{p(D \mid \Lambda)p(\Lambda)}{\int p(D \mid \Lambda)p(\Lambda)d\Lambda} \approx q(\Lambda \mid D)$$

Simpler distribution

Too complex

As the approximation measure, KL divergence can be used

$$q^*(\Lambda) = \arg\min_q KL[q(\Lambda), p(\Lambda \mid D)]$$

$$q(\Lambda) \equiv q(\Lambda \mid D)$$

(For simplicity of notation)

The minimum of the KL divergence is found using variational method

KL and Lower bound

- Minimizing KL is equal to maximizing lower bound

$$KL[q(\Lambda), p(\Lambda \mid D)] = \int q(\Lambda) \log\left(\frac{q(\Lambda)}{p(\Lambda \mid D)}\right) d\Lambda = \int q(\Lambda) \log\left(\frac{q(\Lambda) p(D)}{p(\Lambda, D)}\right) d\Lambda$$

$$= \log(p(D)) + \int q(\Lambda) \log\left(\frac{q(\Lambda)}{p(\Lambda, D)}\right) d\Lambda$$

$$\log p(D) = KL[q(\Lambda), p(\Lambda \mid D)] + \int q(\Lambda) \log\left(\frac{p(\Lambda, D)}{q(\Lambda)}\right) d\Lambda$$

Model evidence
(Constant for q)

Lower bound

$$\mathcal{L}[q] \equiv \int q(\Lambda) \log\left(\frac{p(\Lambda, D)}{q(\Lambda)}\right) d\Lambda$$

Approximation by Factorization

- Assumes (groups of) hidden variables are conditionally independent
  - It is called a mean filed approximation by analogy to physics
  - No restriction on the functional forms

$$p(\Lambda_1, \Lambda_2, \cdots, \Lambda_M \mid D) \approx \prod_{i=1}^{M} q_i(\Lambda_i) \qquad \Lambda = \{\Lambda_1, \Lambda_2, \cdots, \Lambda_M\}$$

Posterior Inference with Mean Field Approximation

- Suppose a probability model consist of two (groups of) hidden variables H, Z and a (group of) observation variable D
- We approximate p(H,Z|D) as q(H,Z|D)=q(H)q(Z)

$$\arg\max_{q(Z)q(H)} \mathcal{L}[q] = \arg\max_{q(Z)q(H)} \mathcal{L}[q(Z)q(H)], \qquad \int q(Z)dZ = 1, \quad \int q(H)dH = 1$$

The method of Lagrange multiplier

$$\arg\max_{q(Z)q(H)} F[q]$$

$$F[q(Z)q(H)] = \mathcal{L}[q(Z)q(H)] - \lambda_Z\left(\int q(Z)dZ - 1\right) - \lambda_H\left(\int q(H)dH - 1\right)$$

Maximization of $F[q]$

$$F[q(Z),q(H)] = \int\int q(Z)q(H)\log\left(\frac{p(Z,H,D)}{q(Z)q(H)}\right)dZdH - \lambda_Z\left(\int q(Z)dZ - 1\right) - \lambda_H\left(\int q(H)dH - 1\right)$$

Variational method
(See the appendix for the derivation)

$$\frac{\partial}{\partial q_Z}\int q_Z q_H \log\left(\frac{p(Z,H,D)}{q_Z q_H}\right)dH - \lambda_Z q_Z = 0$$

$$\frac{\partial}{\partial q_H}\int q_Z q_H \log\left(\frac{p(Z,H,D)}{q_Z q_H}\right)dZ - \lambda_H q_H = 0$$

$$q_z = q(Z), \quad q_H = q(H)$$

$$q(Z) = C_Z \exp\int q(H)\log(p(Z,H,D))dH$$

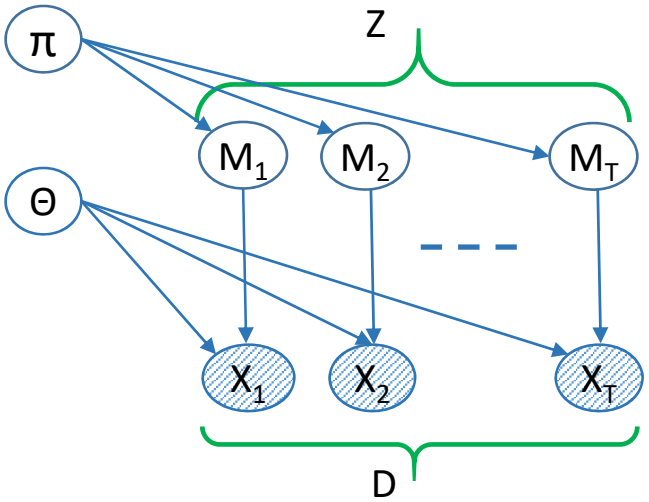
$$q(H) = C_H \exp\int q(Z)\log(p(Z,H,D))dZ$$

C_Z, C_H: normalization constant

The maximum is obtained by alternatively updating $q(Z)$ and $q(H)$ starting from an initial distribution

Variational GMM

$$
\begin{aligned}
p &= P(\pi, \theta, Z, D) \\
&= P(\pi)P(\theta)P(Z \mid \pi)P(D \mid \theta, Z) \\
&= P(\pi)P(\theta)\prod_t P(M_t \mid \pi)P(X_t \mid \theta, M_t)
\end{aligned}
$$



Mean field approximation:

$$P(\pi, \theta, Z \mid D) \approx q(\pi, \theta, Z \mid D) = q(\pi, \theta \mid D)q(Z \mid D)$$

$$q(\pi, \theta) \propto \exp \sum_Z q(Z) \log(p(\pi, \theta, Z, D))$$

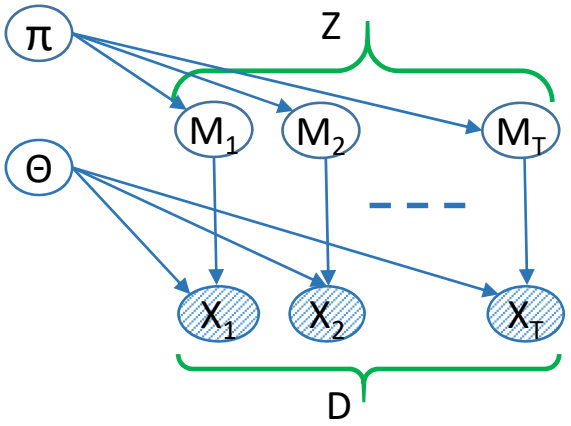
$$q(Z) \propto \exp \int q(\pi, \theta) \log(p(\pi, \theta, Z, D)) d\pi d\theta$$

Cont.

$$q(\pi) \propto P(\pi)\exp\sum_t\sum_{M_t} q(M_t)\log(P(M_t \mid \pi))$$

$$q(\theta) \propto P(\theta)\exp\sum_t\sum_{M_t} q(M_t)\log(P(X_t \mid \theta, M_t))$$

$$q(Z) = \prod_t q(M_t)$$



$$q(M_t) \propto \exp\int\int q(\pi)q(\theta)\log(P(M_t \mid \pi)P(X_t \mid \theta, M_t))d\pi d\theta$$

$$\propto \exp\int q(\pi)\log(P(M_t \mid \pi))d\pi \cdot \exp\int q(\theta)\log(P(X_t \mid \theta, M_t))d\theta$$

Cf. (compare with the above results)

$$P(\pi \mid Z) = \frac{P(\pi)P(Z \mid \pi)}{P(Z)} \propto P(\pi)P(Z \mid \pi) = P(\pi)\prod_t P(M_t \mid \pi) = P(\pi)\exp\sum_t \log P(M_t \mid \pi)$$

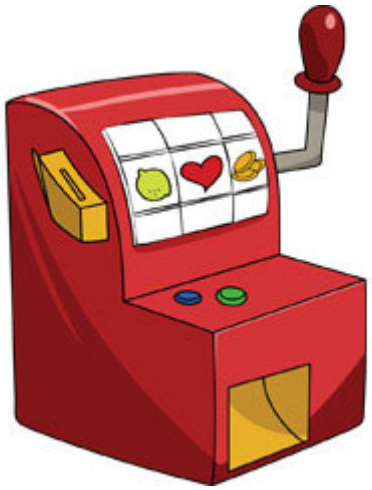
$$P(\theta \mid Z, D) \propto P(\theta)P(D \mid \theta, Z) = P(\theta)\exp\sum_t \log P(X_t \mid \theta, M_t)$$

$$P(M_t \mid \theta, \pi, X_t) \propto P(M_t \mid \pi)P(X_t \mid \theta, M_t) = \exp(\log(P(M_t \mid \pi))) \cdot \exp(\log(P(X_t \mid \theta, M_t)))$$

Sampling Methods

Pseudo Random Generator

- On digital computer, everything is deterministically calculated and there is no randomness
- However, sometimes we want random numbers
- Most programing languages have a pseudo random generator function



```
Python 2.6
> import random
> random.random()
0.8938901900395423
> random.random()
0.9856359157198963 9
> random.random()
0.5305444355684372
```

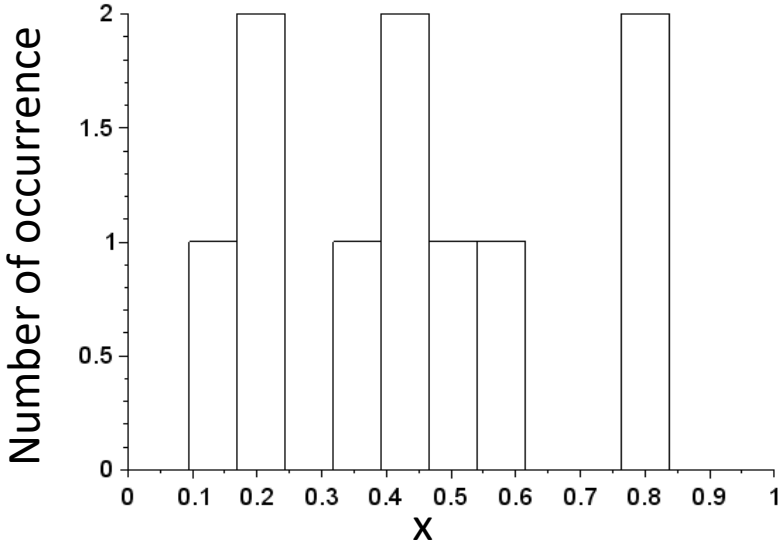
Sampling From a Uniform Distribution

- Random numbers distributed uniformly over some region

Example:

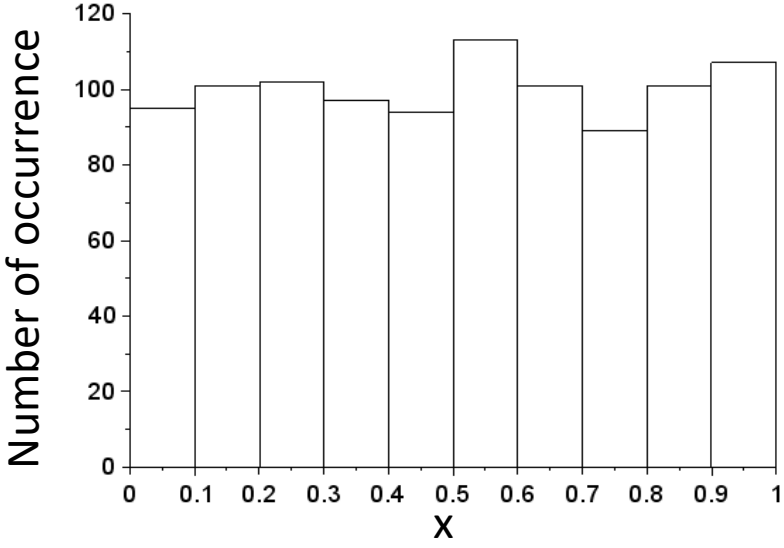
Histogram of samples obtained from a uniform distribution over (0, 1)
(To make the graph, scilab was used)

10 samples



histplot(10,rand(1:10),
normalization=%f)

1000 samples



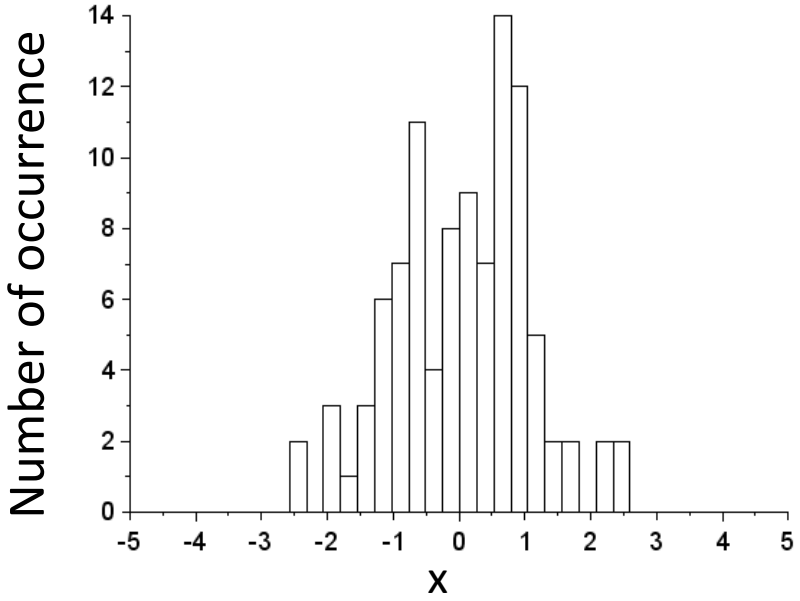
histplot(10,rand(1:1000),
normalization=%f)

Sampling From a Gaussian Distribution

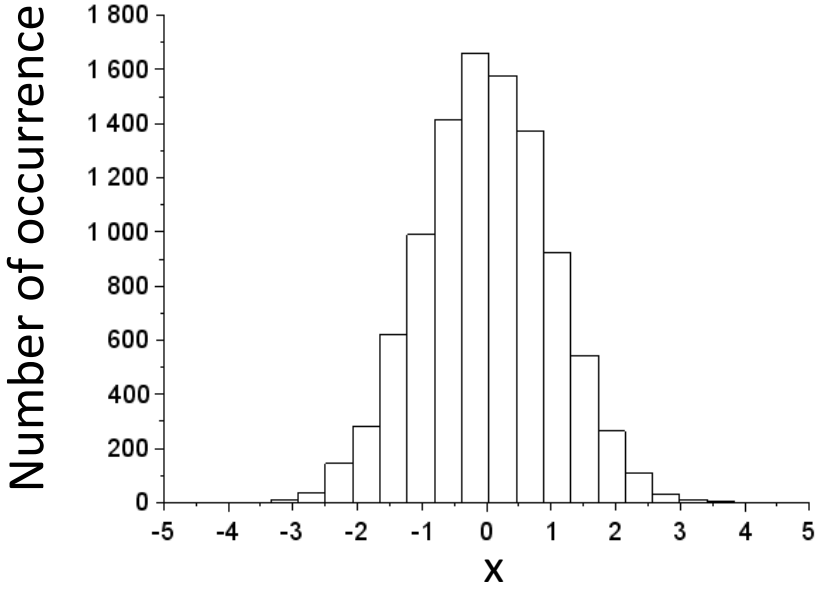
- Standard normal (Gaussian) distribution has a mean 0.0 and a variance 1.0

Example:

100 samples



10000 samples



Transform of Random Variable

- Let x be a random variable and f be a function $y = f(x)$. When x follows $p(x)$, y follows the following distribution $q(y)$

$$q(y) = p(x)\left|\frac{dx}{dy}\right|$$

Example

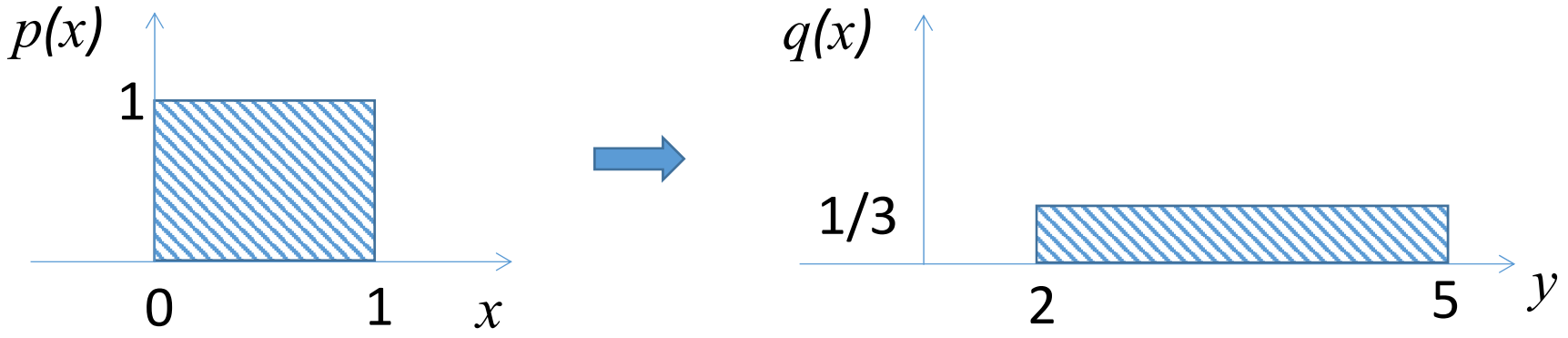
- When $p(x)$ and $y = f(x)$ are given as follows, obtain distribution $q(y)$

$$p(x) = 1 \quad x \in (0, 1)$$

$$y = 3x + 2$$

Answer

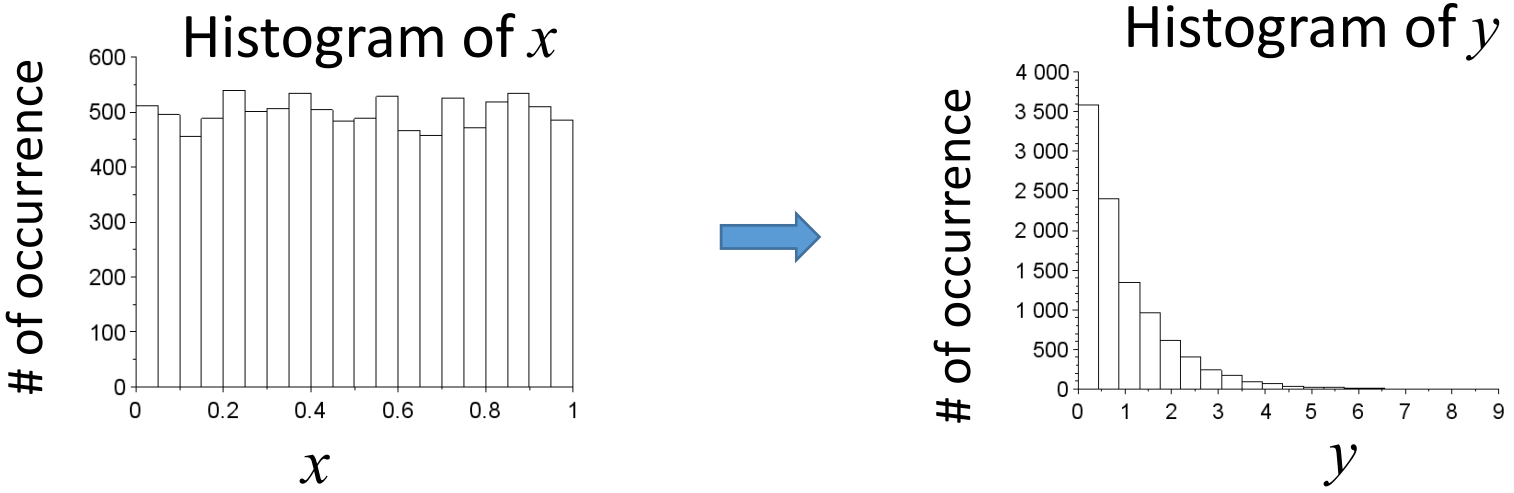
$$x = \frac{y-2}{3}, \qquad \left|\frac{dx}{dy}\right| = \frac{1}{3} \qquad \Rightarrow \qquad q(y) = \frac{1}{3} \quad x \in (2, 5)$$



Exercise 4.1

- When $p(x)$ and $y = f(x)$ are given as follows, obtain distribution $q(y)$

$$p(x) = 1 \quad x \in (0, 1), \qquad y = -\log(1 - x)$$



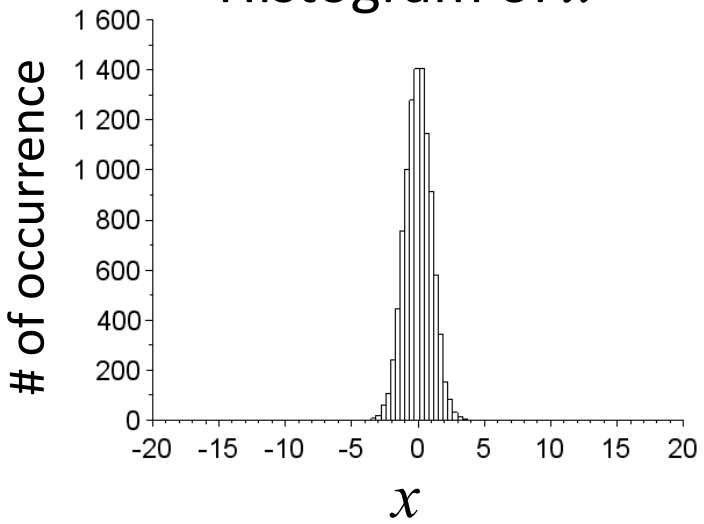
Exercise 4.2

- When $p(x)$ and $y = f(x)$ are given as follows, obtain distribution q(y)

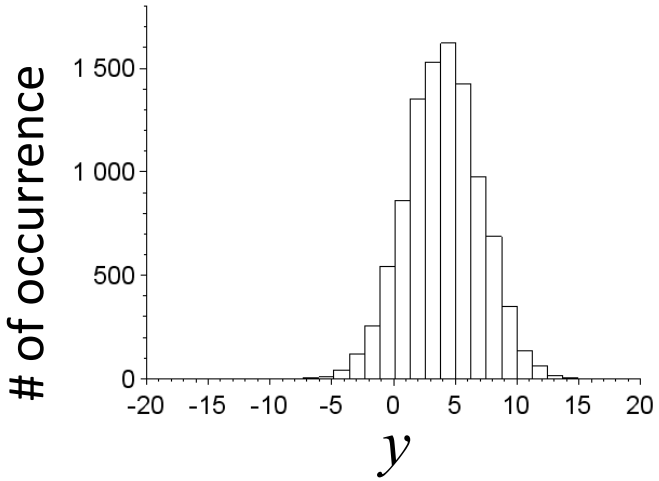
$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) = N(x \mid 0,1) \quad x \in (-\infty, \infty), \qquad y = 3x + 4$$

The answer becomes a Gaussian distribution. Report its mean and variance.

Histogram of x



Histogram of y

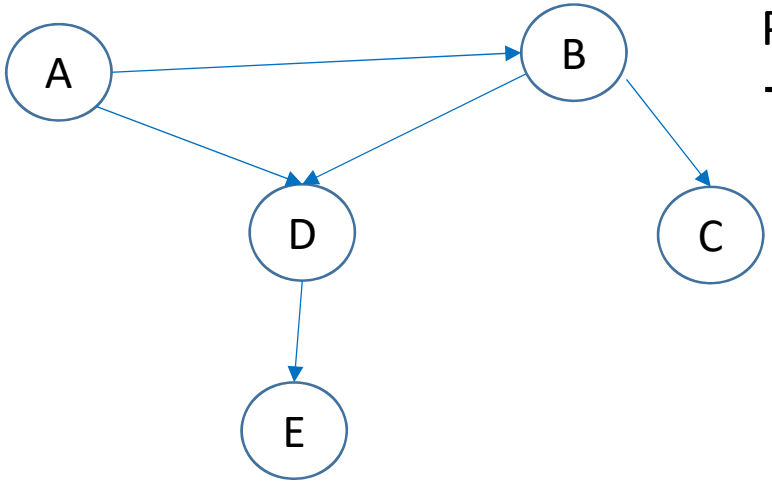


Sampling form Complex Distributions

- Distributions that can be obtained by a transformation from a simple distribution (such as uniform distribution) is limited
- We need sampling methods that do not require integral and inverse of a function, and can be applied to more complex distributions
  - Ancestral sampling
  - Rejection sampling
  - Markov Chain Monte Carlo (MCMC)

Ancestral Sampling

- Assumptions:
  - We want samples from a joint distribution p(x_1, x_2,…, x_M)
  - The joint distribution is given as a Bayesian network
  - Sampling from the conditional distributions are easy
- Algorithm:
  - Sample from the parent nodes to the child nodes in order
  - For the child nodes, use the already sampled parent value in the conditional part

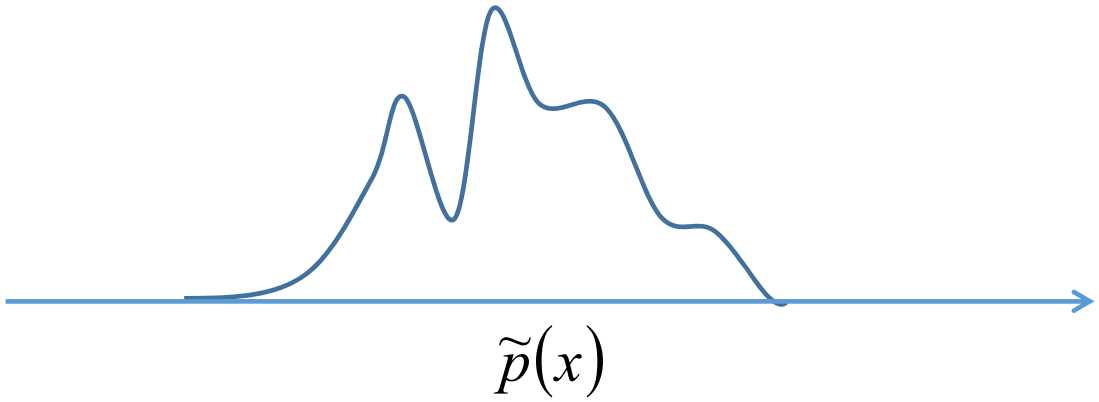


P(A) → P(B|A) → P(C|B)
→ P(D|A,B) → P(E|D)

Rejection Sampling

- Assumptions:
  - We want samples from a distribution $p(x)$. The normalization constant Z may be unknown.

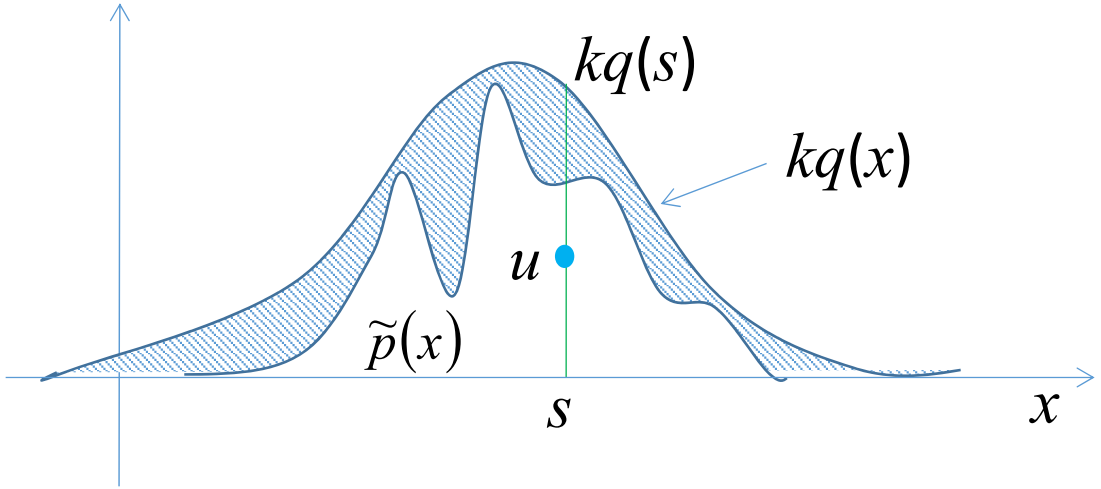
$$p(x) = \frac{1}{Z}\tilde{p}(x)$$



  - We have a distribution $q(x)$ from which we can easily derive samples. We refer to q as a proposal distraction

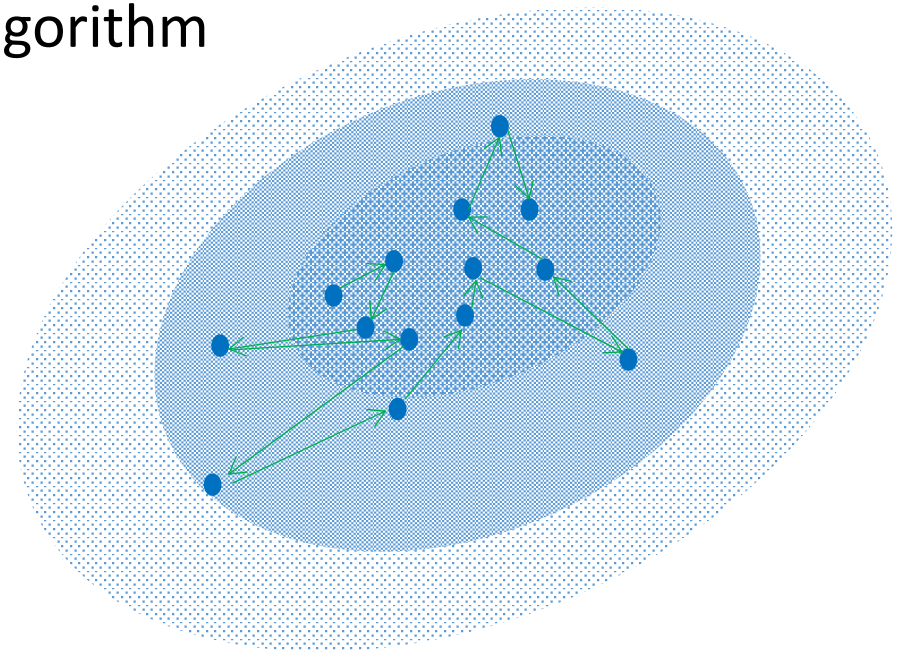
Procedure of Rejection Sampling

- Algorithm:
  1. Choose a constant *k* so that the following holds
  $$kq(x) \geq \tilde{p}(x)$$
  2. Derive a sample s from $q(x)$
  3. Derive a sample u from a uniform distribution ranging [0, $kq(s)$]
  4. If $u > \tilde{p}(x)$ then reject the sample. Otherwise, adopt it



Markov Chain Monte Carlo

- General and powerful framework for sampling
  - Scales well with the dimensionality of the sample space
- Maintains a state that forms a Markov chain. The set of the states follows the desired distribution
  - Metropolis algorithm
  - Metropolis-Hastings algorithm
  - Gibbs Sampling



Metropolis Algorithm

- Assumptions:
  - We want samples from a distribution $p(X)$

$$p(X) = \frac{1}{Z}\tilde{p}(X)$$

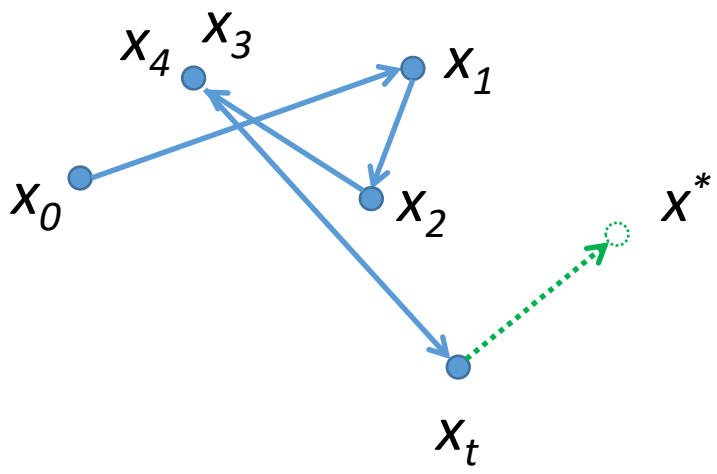
  - The normalization constant Z may be unknown
- Initialization:
  1. Prepare a symmetric proposal distribution $q(x_A|x_B)$ that satisfy $q(x_A|x_B)=q(x_B|x_A)$
  2. Prepare an initial state x_0

Exercise 4.3

- Show that $N(x_A|x_B, 1) = N(x_B|x_A,1)$, where $N(x|m,v)$ is the Gaussian distribution with mean m and variance v

Procedure of Metropolis Algorithm

- Algorithm:
  1. Get a candidate sample x^* from the proposal distribution $q(x|x_t)$ based on the current state x_t
  2. Accept the candidate with probability $A(x^*, x_t) = \min\left(1, \frac{\tilde{p}(x^*)}{\tilde{p}(x_t)}\right)$ or discard it
  3. If the candidate is accepted, save it as the next state x_{t+1}. If it is discarded, then set x_{t+1} equals to x_t
  4. Goto step 3



Metropolis-Hastings Algorithm

- An extension of the Metropolis algorithm
- No symmetric requirement for the proposal distribution
- The acceptance probability of the candidate state is defined as follows. Other part is the same as the Metropolis algorithm

$$A_k(x^*, x_t) = \min\left(1, \frac{\tilde{p}(x^*) q_k(x_t \mid x^*)}{\tilde{p}(x_t) q_k(x^* \mid x_t)}\right)$$

Gibbs Sampling

- Problem:
  - We want samples from a joint distribution p(x_1, x_2,…, x_M)
- Algorithm:
  1. Prepare an initial state $X_0 = <x_1,x_2,\ldots,x_M>_0$
  2. Select one of the variables x_i in order or at random
  3. Get a sample from $p(x_i \mid X \setminus i)$ and update x_i with that value
  4. Goto step 2. After enough iterations, the distribution of x_t follows p(X)

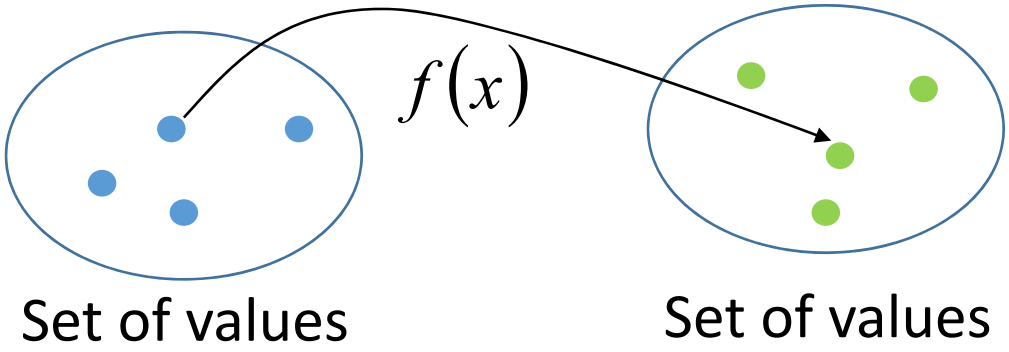
Compared to the Metropolis algorithm:

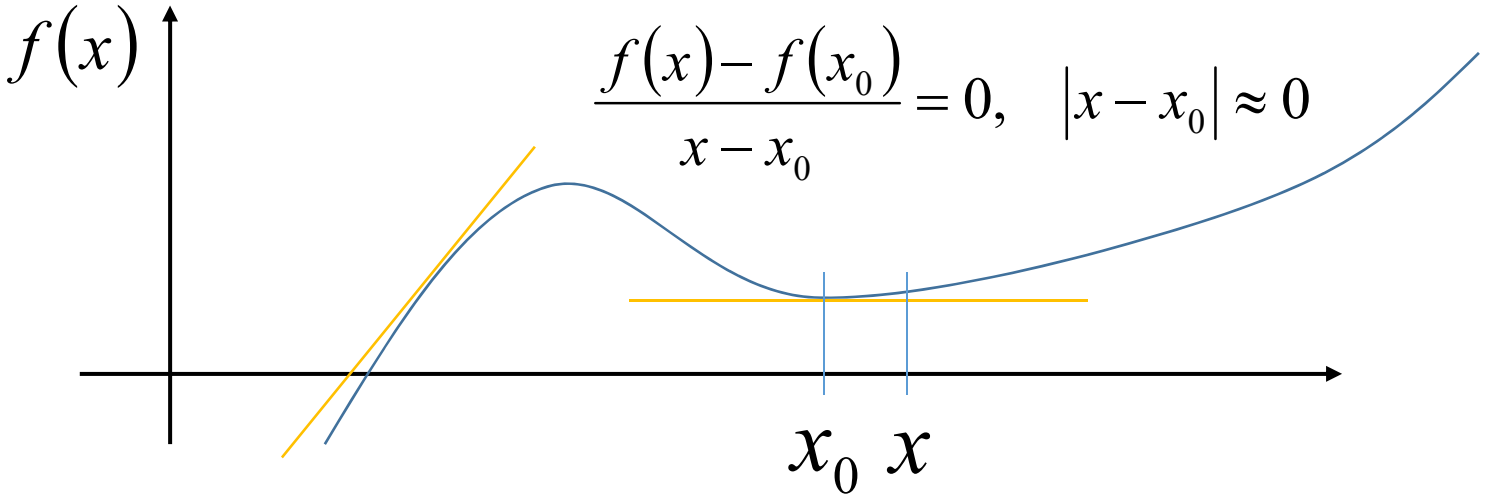
- Sampling from conditional distribution of x_i given all other variables need to be feasible
- There is no rejection step

Appendix

Variational Method (Outline)

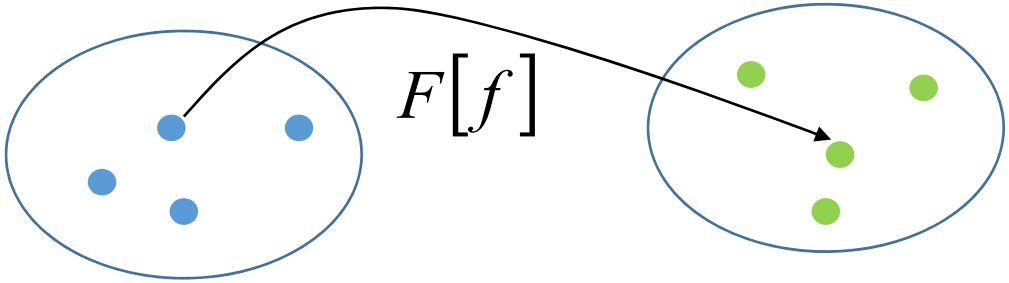
- Review of derivatives
  - Function: a mapping from a value to a value





Functional and Functional Derivative

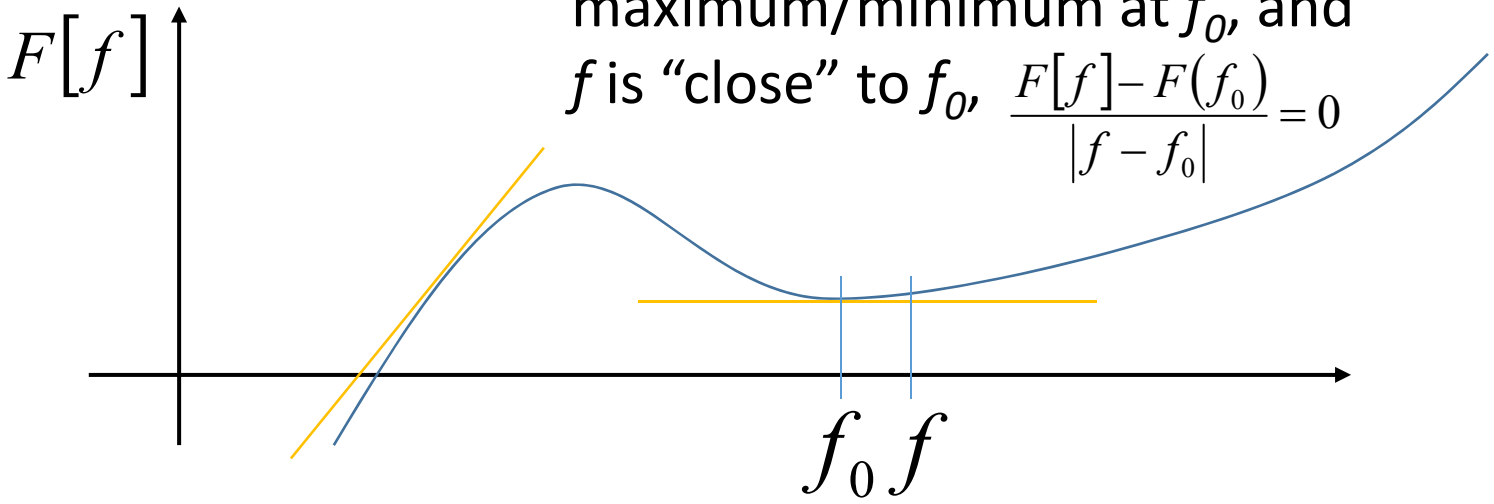
- Functional



Set of functions Set of values

Ex. Entropy H[p] takes a function p (probability distribution) and returns a value

If a functional *F* takes a maximum/minimum at f_0, and *f* is "close" to f_0, $\frac{F[f]-F(f_0)}{|f-f_0|}=0$



Euler-Lagrange Equation

$F[f]$

F is a functional of *f*

$f(x) = f_0(x) + \varepsilon\eta(x)$

Suppose *F* takes minimum/maximum at f_0 . Let η be an arbitral function of *x*, and ε is a scalar constant

$g(\varepsilon) = F[f]$

g(ε) is a function of ε
(takes and returns a scalar)

When ε is closed to 0, *f* is close to f_0.

Therefore, $\frac{F[f] - F[f_0]}{\varepsilon - 0} = 0$.

This must hold for arbitral η.

$$\left.\frac{\partial}{\partial\varepsilon} F[f]\right|_{\varepsilon=0} = \left.\frac{\partial}{\partial\varepsilon} g(\varepsilon)\right|_{\varepsilon=0} = 0$$

Merely a derivative of a function

Cont.

- When $F[f]=\int h(f)dx+C,$

$$\frac{\partial}{\partial\varepsilon}F[f]=\int\frac{\partial}{\partial\varepsilon}h(f(x))dx=\int\frac{\partial h}{\partial f}\frac{\partial f}{\partial\varepsilon}dx=\int\frac{\partial h}{\partial f}\eta(x)dx$$

$$\Rightarrow \int\frac{\partial h}{\partial f}\eta(x)dx=0$$ must hold for arbitral η.

$$\Rightarrow \frac{\partial h}{\partial f}=0$$

C.f. How about when $F[f]=\int_a^b h(f,f',x)dx,\quad \eta(a)=\eta(b)=0$