电子科技大学
University of Electronic Science and Technology of China

Particle-Based Approximate Inference

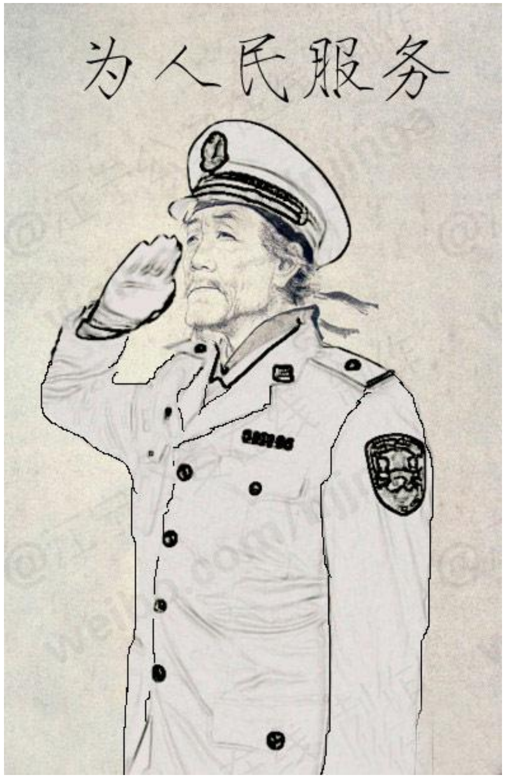
杨小麟&韩葳

DM LESS IS MORE

Data Mining Lab, Big Data Research Center, UESTC
School of Computer Science and Engineering
Email：xiaolinyn@gmail.com

寄语

- 临渊羡鱼，不如退而结网。——《汉书》
- 当局者迷，旁观者清。——《旧唐书》



Outline

数据挖掘实验室
Data Mining Lab

- ✓ Motivation
- ✓ Some Basic Methods
  1. Naive Sampling/Forward Sampling
  2. Rejection Sampling
  3. Importance Sampling
     3.1 Unnormalized Importance Sampling
     3.2 Normalized Importance Sampling
        3.2.1 Likelihood Weighting
- ✓ MCMC(Markov Chain Monte Carlo)
  1. MCMC
  2. Gibbs Sampling

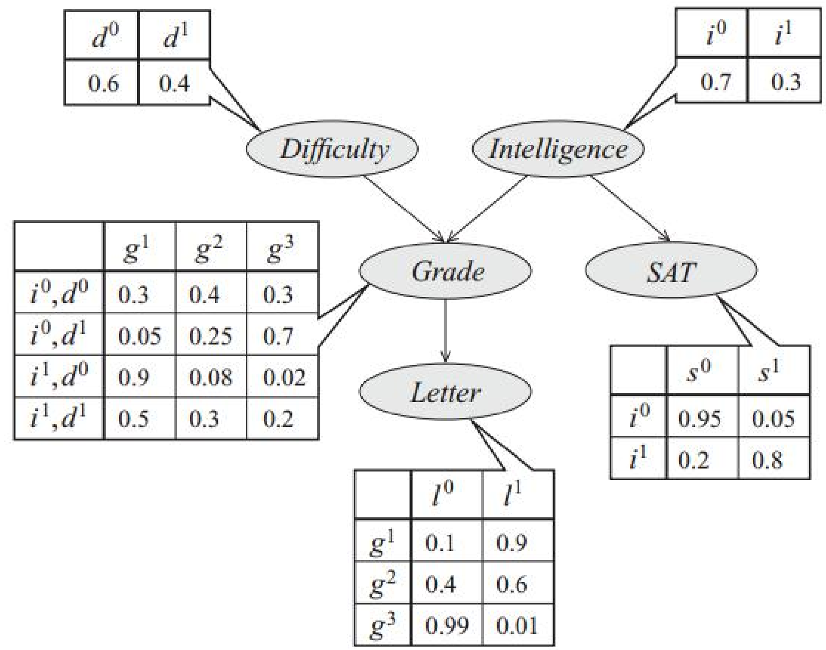
01 Motivation

Describe our world!

- Representation, Learning , Inference

Inference: response to queries (what we want to know)

- $P(z|e)$
- $E(f(z))$



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

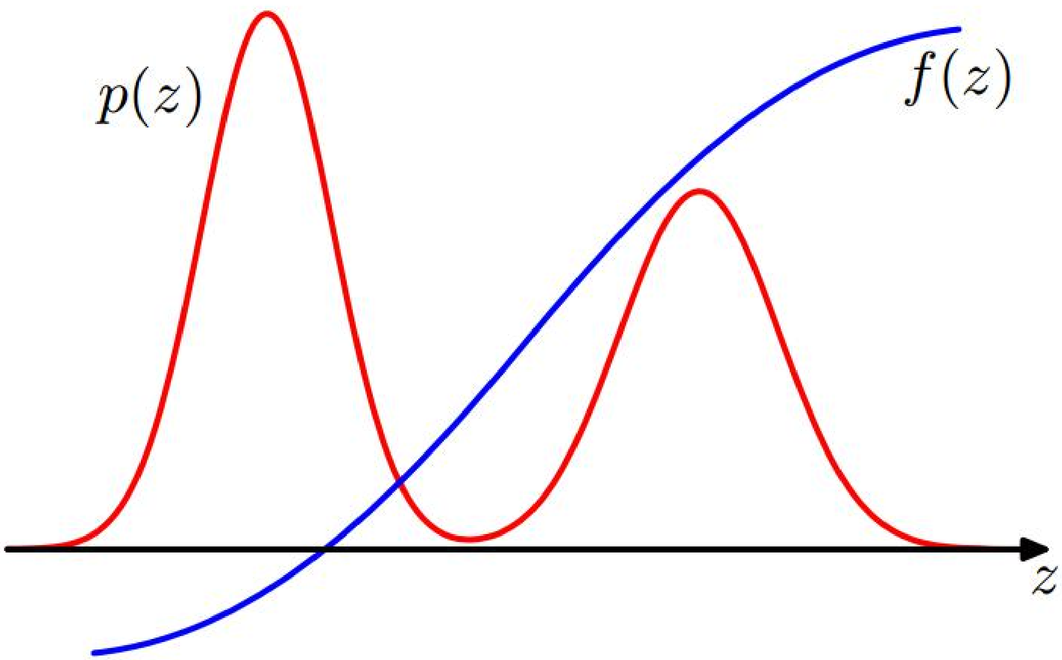
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Figure 1: Student Bayesian network $\boldsymbol{B}_{student}$ with CPDs

Motivation (王婆卖瓜)

From a high level, it appears that **sampling methods are the ultimate general-purpose inference algorithm**. They are the **only** method that can be applied to **arbitrary probabilistic models** and that is guaranteed to **achieve the correct results at the large sample limit**.



Motivation

- How to sample?

  In most case, we want to find $E(f(z)) = \int f(z)p(z)dz$

  - We draw samples $\{z^i\}_M$ from f(z) i.i.d.

  And we set $\boxed{E(\hat{f}(z)) = \frac{1}{M}\sum_{i=1}^{M} f(z^i)}$

$$\Downarrow$$

$$\hat{P}_D(z) = \frac{1}{M}\sum_{i=1}^{M} I(z^m = z)$$

Motivation

- Why sampling work?
  - Hoeffding bound:

$$P_D(\hat{P}_D(z) \notin [P(z) - \varepsilon, P(z) + \varepsilon]) \leq 2e^{-2M\varepsilon^2} \leq \delta$$

$$\Downarrow$$

$$M \geq \frac{\ln(2/\delta)}{2\varepsilon^2}$$

Motivation

- Common transformation:

Target: generate random numbers from simple nonuniform distribution.

- Method(Transformation technique):

given $z \sim Uniform(0,1)$ using some function $f(\cdot)$ to transform z so that $y = f(z)$

➡ the distribution of y is: $p(y) = p(z)\left|\frac{dz}{dy}\right|$

➡ $p(z) = 1$ $\quad z = h(y) = \int_{-\infty}^{y} p(\hat{y})\,d\hat{y}$

➡ $y = h^{-1}(z) = f(z)$

Motivation

- One example:

draw samples from exponential distribution

$$p(y) = \lambda \exp(-\lambda y) \quad 0 \leq \lambda < \infty$$

$$\Rightarrow \quad h(y) = 1 - \exp(-\lambda y)$$

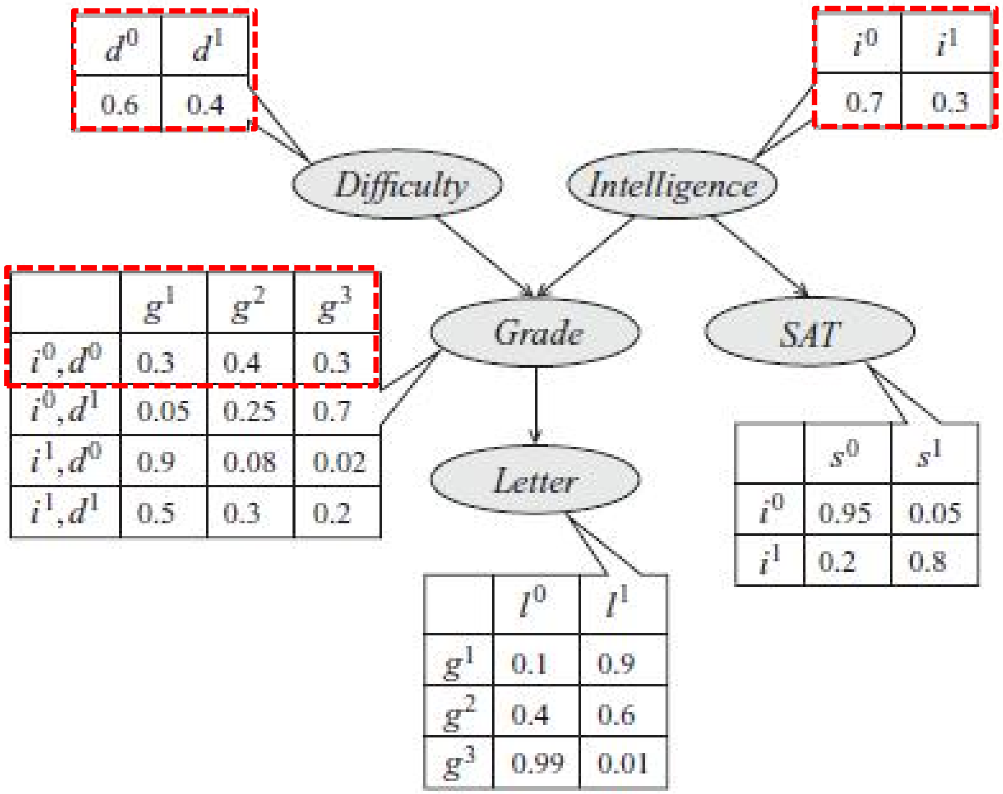
$$\Rightarrow \quad y = -\lambda^{-1} \ln(1 - z)$$

Thus, if we transform the uniformly distributed variable z using $y = -\lambda^{-1}\ln(1-z)$, then y will have an exponential distribution.

2 Some Basic Methods

Naive Sampling/Forward Sampling

- In the BN:



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

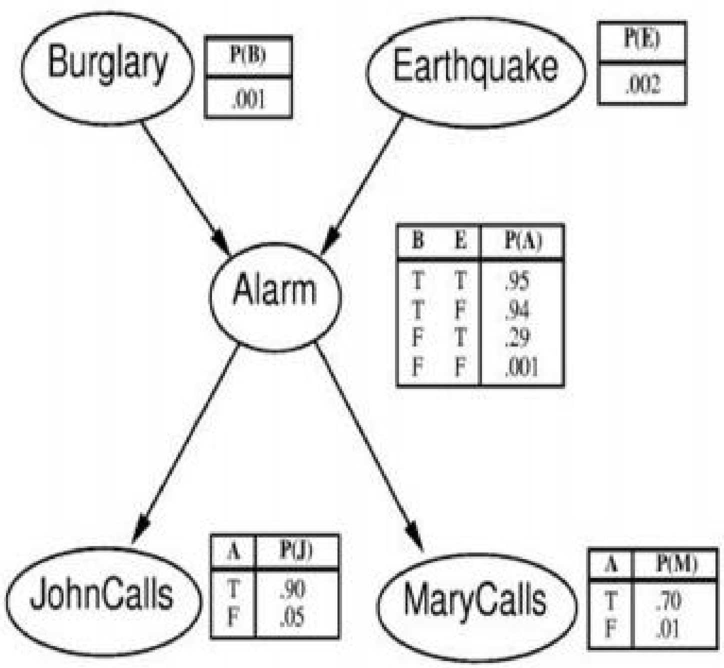
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- First, figuratively toss a coin to draw the sample from D. Assume we get d^0.
- Similarly, toss a coin to draw the sample from I. Assume we get i^0.
- Then we sample for G given d^0 and i^0.
- The process continues similarly for S and L.

Naive Sampling/Forward Sampling

- The problems:
  - Apply (at least in their simple form) only to Bayesian networks, in undirected models, even generating a sample from the prior distribution is a difficult task.
  - For a model with hundreds or more variables, rare events will be very hard to gain enough samples even after a long time for sampling.



E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

$P(J|B1)=P(J,B1)/P(B1)$

can not defined

Rejection Sampling

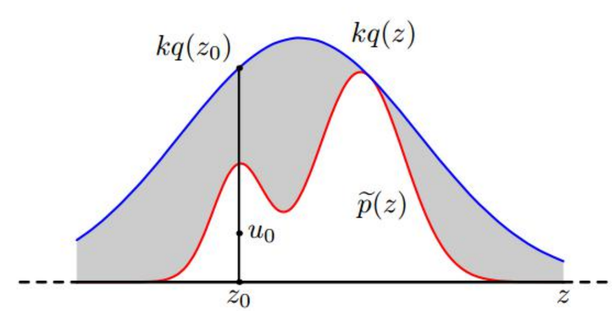
- Suppose we wish to sample from :

$$p(x) = p'(x)/\alpha$$

- It's common that $p(x)$ is difficult to sample or even to compute, But $p'(x)$ is easy to evaluate. $p(x)$ is often called the **target distribution**.

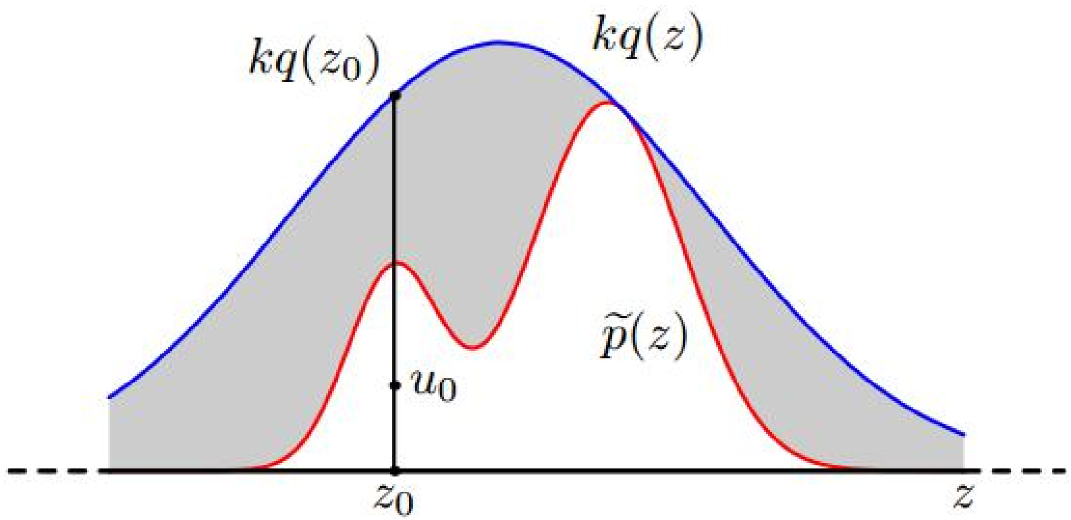
- Sample from a simple distribution $q(x)$, known as the **proposal distribution.**

- Introduce a constant k who guarantees that $kq(z) \geq \tilde{p}(z)$ for all values of z.



Rejection Sampling

- Three steps for each sample:
  - First, generate a number z_0 from the distribution $q(z)$.
  - Next, generate a number u_0 from the uniform distribution over $[0, kq(z_0)]$.
  - Finally, if $u_0 > \tilde{p}(z_0)$ then the sample is rejected, otherwise z_0 is retained.



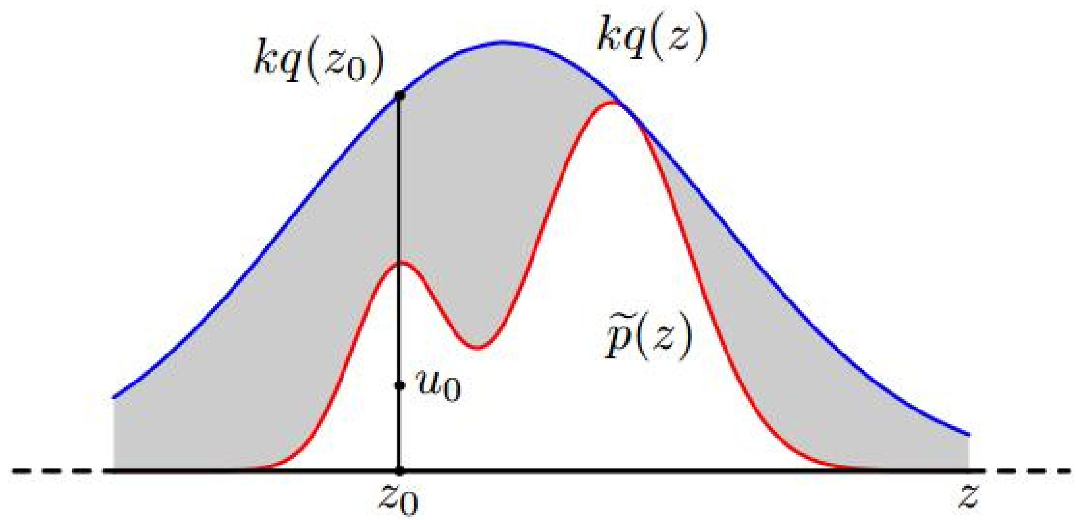
Rejection Sampling

- Correctness:

$$p_D(z) \propto \frac{[p'(z)/k\,q(z)]q(z)}{\int [p'(z)/k\,q(z)]q(z)dx}$$

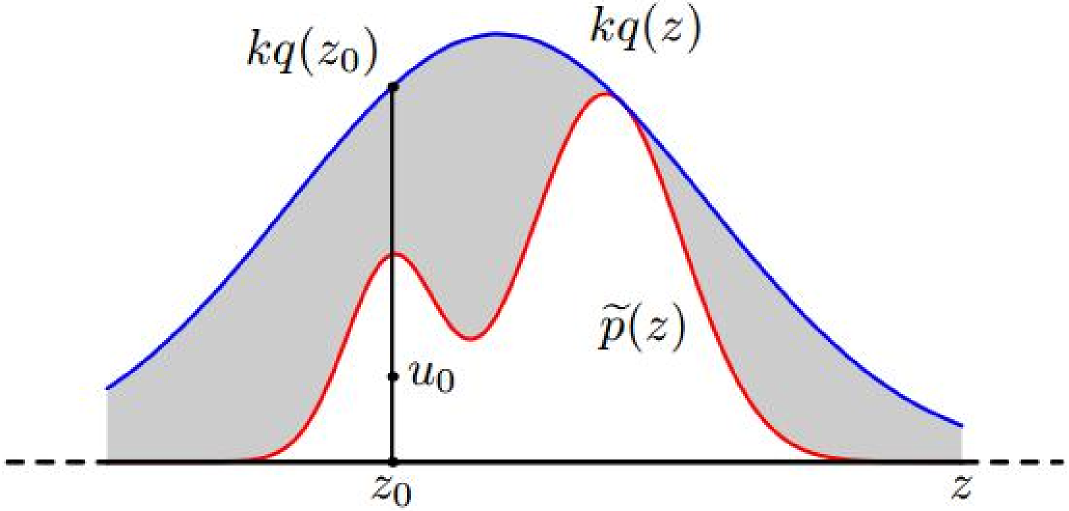
$$= \frac{p'(z)}{\int p'(z)dx}$$

$$= p(z)$$



Rejection Sampling

- Drawbacks:
  - In low dimensions, the shape of $p'(x)$ and $q(x)$ need to be similar with each other, otherwise we will reject lots of samples.
  - In high dimensions, even if the shapes of these two distributions are similar, the rejection rate is really high.(leave out the proof)



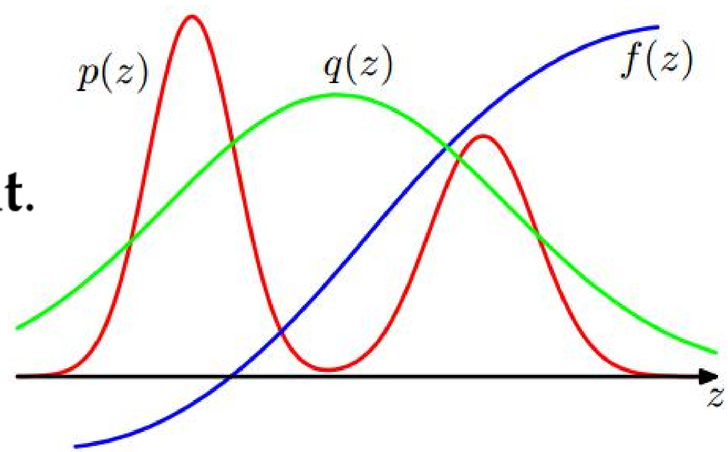
Unnormalized Importance Sampling

- Suppose sampling from $P(x)$ is hard, but we can sample from a simpler proposal distribution $Q(x)$.

- If Q dominates P (i.e., $Q(x) > 0$ whenever $P(x) > 0$), the procedure to compute $E(f(x))$ is:

  - Sample $x^m \sim Q(x)$ for m = 1, 2, 3, …, M
  - Compute $\hat{f} = \frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P(x^m)}{Q(x^m)}$

$\frac{P(x^m)}{Q(x^m)}$ is known as **importance weight**.

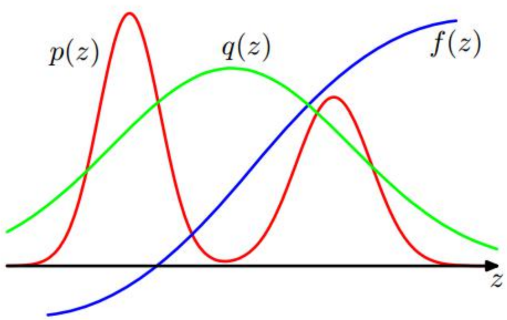


Unnormalized Importance Sampling

数据挖掘实验室
Data Mining Lab

- Claim: $\hat{f}$ is an unbiased estimator of $E_P(f(x))$:

$$\begin{aligned}
\mathbb{E}_Q[\hat{f}] &= \mathbb{E}_Q[\frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P(x^m)}{Q(x^m)}] \\
&= \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_Q[f(x^m)\frac{P(x^m)}{Q(x^m)}] \\
&= \mathbb{E}_{x\sim Q}[f(x)\frac{P(x)}{Q(x)}] \qquad \text{as } x^m \text{ are i.i.d drawn from } Q \\
&= \int f(x)\frac{P(x)}{Q(x)}Q(x)\,dx \\
&= \int f(x)P(x)\,dx \\
&= \mathbb{E}_P[f(x)]
\end{aligned}$$



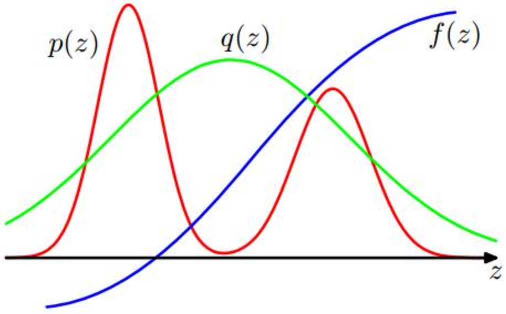
Normalized Importance Sampling

- Suppose we can only evaluate $P'(x) = \alpha P(x)$ for some unknown scaling factor $\alpha > 0$.(e.g. for an MRF)
- We can eliminate the nasty normalization constant α as follows:
  - Let $r(x) = \dfrac{P'(x)}{Q(x)}$

$$\Rightarrow \quad \mathbb{E}_Q[r(x)] = \mathbb{E}_Q[\frac{P'(x)}{Q(x)}] = \underline{\int \frac{P'(x)}{Q(x)} Q(x)\, dx} = \int P'(x)\, dx = \alpha$$

$$\Downarrow$$

$$\hat{\alpha} = \frac{1}{M} \sum_{m=1}^{M} r(x^m)$$

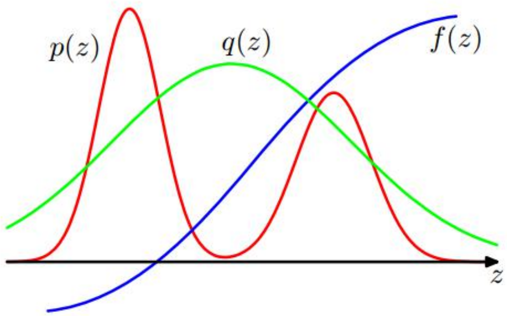


Normalized Importance Sampling

数据挖掘实验室
Data Mining Lab

- The procedure to compute E(f(x)) is:
  - Sample $x^m \sim Q(x)$ for m = 1, 2, 3, …, M
  - Compute scaling factor estimator $\hat{\alpha} = \frac{1}{M}\sum_{m=1}^{M} r(x^m)$
  - Compute

$$\widehat{f} = \frac{1}{\widehat{\alpha}}\frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P'(x^m)}{Q(x^m)} = \frac{\sum_{m=1}^{M} f(x^m) r(x^m)}{\sum_{m=1}^{M} r(x^m)}$$



Normalized Importance Sampling

- Correctness:

$$P(x) = \frac{P'(x)}{\alpha}$$

$$E_P(f(x)) = \int f(x)P(x)dx = \frac{1}{\alpha}\int f(x)\frac{P'(x)}{Q(x)}Q(x)dx$$

$$= \frac{\int f(x)\boxed{r(x)}Q(x)dx}{\boxed{\int r(x)Q(x)dx}}$$

$$r(x) = \frac{P'(x)}{Q(x)}$$

$$\alpha = \int r(x)Q(x)dx$$

$$\frac{1}{M}$$

$$\approx \frac{\sum_m f(x^m)\, r^m}{\sum_m r^m} \qquad where \;\; x^m \sim Q(x)$$

$$= \sum_m f(x^m) w^m \qquad where \;\; w^m = \frac{r^m}{\sum_m r^m}$$

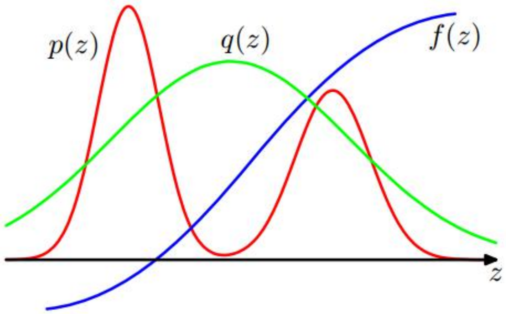
Normalized Importance Sampling

- Claim: Normalized importance sampling is biased.

To show this, suppose we sampled only once, that is, M = 1:

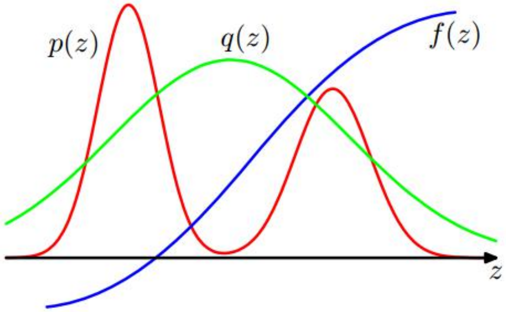
$$\widehat{f} = \frac{f(x_1)r(x_1)}{r(x_1)} = f(x_1)$$

$$\mathbb{E}_Q[\widehat{f}] = \mathbb{E}_Q[f(x_1)] \neq \mathbb{E}_P[f(x_1)] \text{ in general}$$



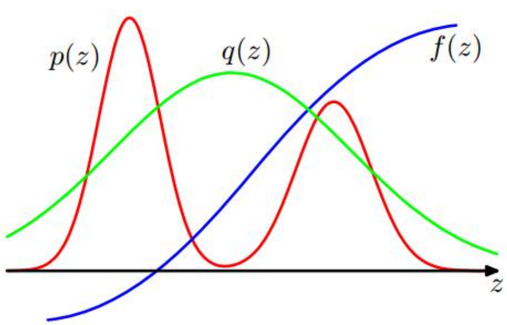
Unnormalized vs Normalized Importance Sampling

- Bias: Unnormalized importance sampling is unbiased, but normalized importance sampling is biased.
- Variance: in practice, the variance of the estimator in the unnormalized case is usually higher than that in the normalized case.
- Requiremet : Unnormalized importance sampling need to calculate $P(x)$, however, it is common to have $P'(x)$ available instead of $P(x)$.



Drawback of Importance Sampling

- **The success of this approach depends crucially on how well the sampling distribution $\boldsymbol{Q(x)}$ matches the desired distribution $\boldsymbol{P(x)}$.** As is often the case, $P(x)f(x)$ is strongly varying and has a significant proportion of its mass concentrated over relatively small regions of x space, then the set of importance weights may be dominated by a few weights having large values, with the remaining weights being relatively insignificant. **Thus the effective sample size can be much smaller than the apparent sample size $\boldsymbol{M}$.**



Likelihood Weighting

- Normalized importance sampling is applied in the Bayes net.
  - The proposal distribution $Q(x)$ (suppose we have gotten the evidence e = {I = i^1 and G = g^2}):

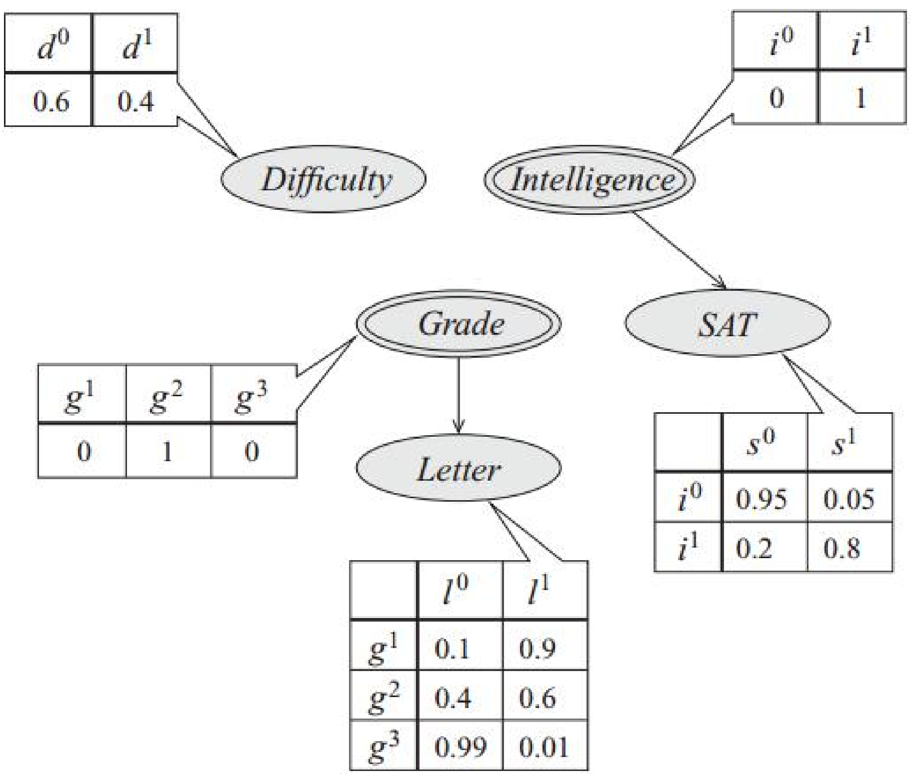


Figure 2: the mutilated(多片段) network

- Define $Q(x) = P_M(x)$ to be the density of the mutilated network.

$$P(D,S,L|e) = \frac{P(D,S,L,e)}{P(e)}$$

But sometimes it's difficult to evaluate $P(e)$.

- Define $P'(x) = P(x,e)$ so that

$$P(x|e) = \frac{P(x,e)}{P(e)} = \frac{P'(x)}{P(e)}$$

- Based on the idea of normalized importance sampling, compute:

$$\hat{P}(X_i = x_i|e) = \frac{\sum_{m=1}^{M} I(x_i^m) r(x^m)}{\sum_{m=1}^{M} r(x^m)}$$

where $r(x^m) = \dfrac{P'(x^m)}{P_M(x^m)}$

Likelihood Weighting

- how to calculate $r(x^m) = \dfrac{P'(x^m)}{P_M(x^m)}$?

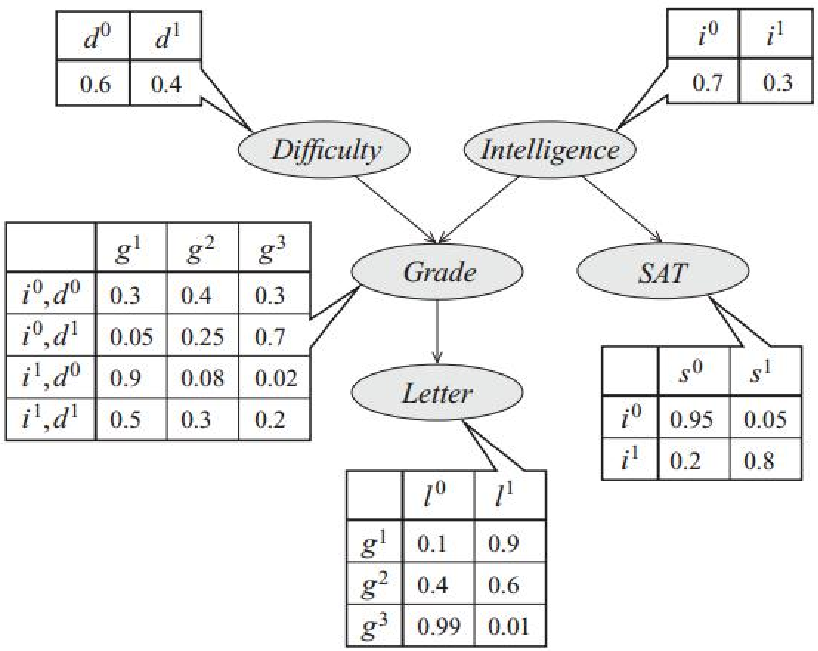


Figure 3: the original network

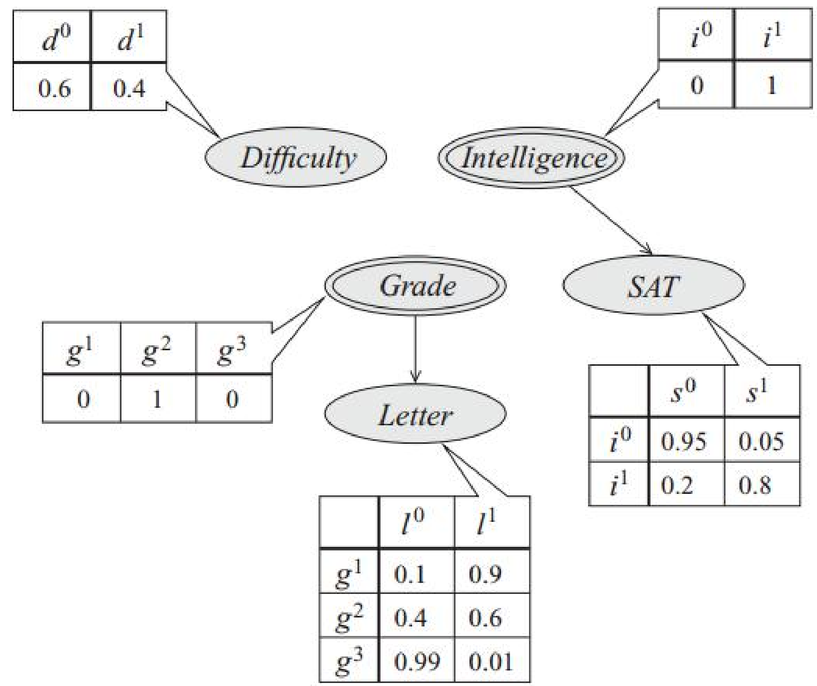


Figure 4: the mutilated network

Likelihood Weighting

$$r(x^m) = \boxed{P'(x^m)} / \boxed{P_M(x^m)}$$

$$P(D, I, G, S, L) = P(D)P(I)P(G \mid D, I)P(S \mid L)P(L \mid G)$$

$$P(D, S, L) = P(D)P(S \mid I)P(L \mid G)$$

Thus $$r(x^m) = P(I)P(G \mid D, I)$$

Algorithm 12.2 Likelihood-weighted particle generation)

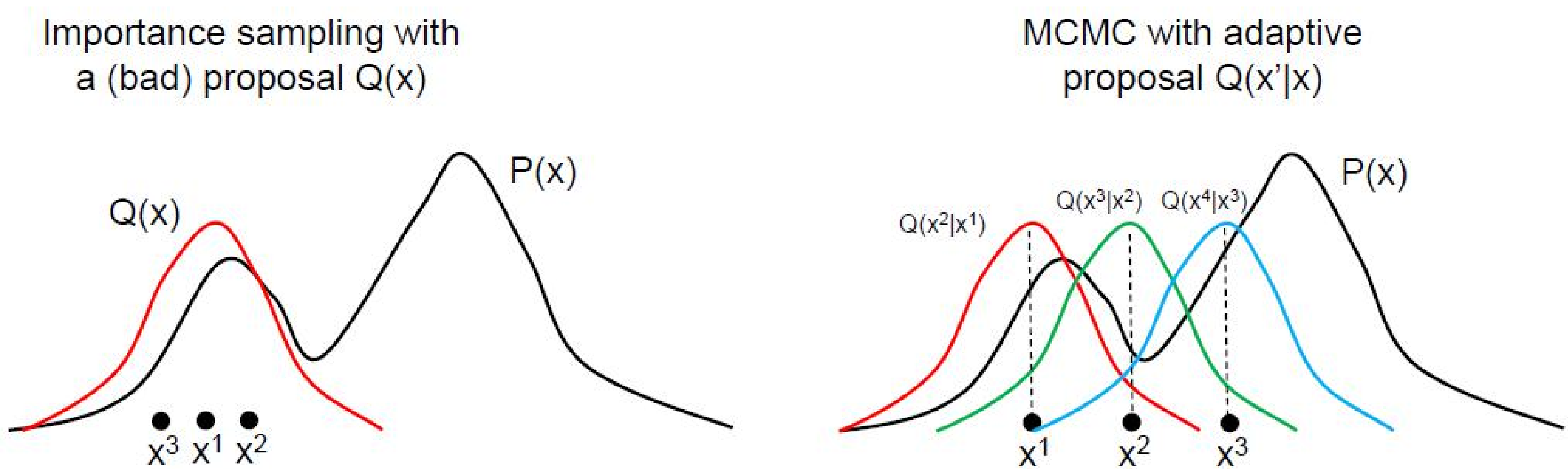
Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $\boldsymbol{Z} = \boldsymbol{z}$ // Event in the network
)
  Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
  $\boxed{w \leftarrow 1}$
  for $i = 1, \ldots, n$
    $\boldsymbol{u}_i \leftarrow \boldsymbol{x}\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
    if $X_i \notin \boldsymbol{Z}$ **then**
      Sample x_i from $P(X_i \mid \boldsymbol{u}_i)$
    else
      $x_i \leftarrow \boldsymbol{z}\langle X_i \rangle$ // Assignment to X_i in $\boldsymbol{z}$
      $\boxed{w \leftarrow w \cdot P(x_i \mid \boldsymbol{u}_i)}$ // Multiply weight by probability of desired value
  return $(x_1, \ldots, x_n), w$

LW indicates that <u>the weights of different samples are derived from the likelihood of the evidence accumulated throughout the sampling process</u>.

不同样本的权重来自于采样过程中累积的证据的似然

03 MCMC

- Intuition:
  - Instead of $Q(x)$, we use $Q(x'|x)$ where x' is the new state being sampled, and x is the previous sample.
  - As x changes, $Q(x'|x)$ can also change(as a function of x').



- Metropolis-Hastings(MH) Algorithm:

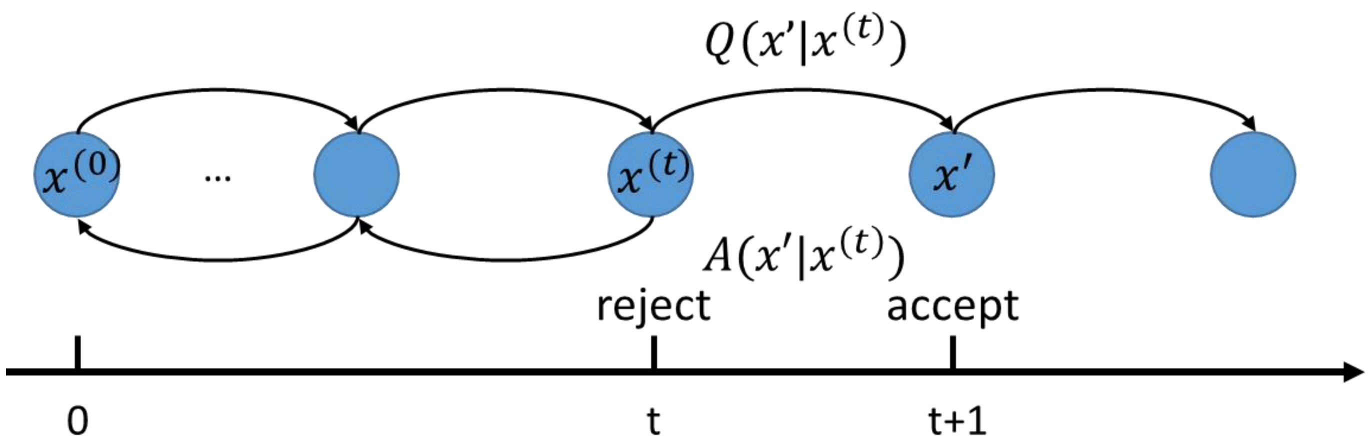
1. Initialize the starting state $x^{(t)}$ at $t = 0$.

2. Draws a sample x' from the proposal $Q(x'|x^{(t)})$. Note that this proposal is now a function of the previously drawn sample $x^{(t)}$ (at time step t).

3. The new sample x' is accepted with the probability:

$$A(x'|x^{(t)}) = \min(1, \frac{P(x')Q(x^{(t)}|x')}{P(x^{(t)})Q(x'|x^{(t)})})$$

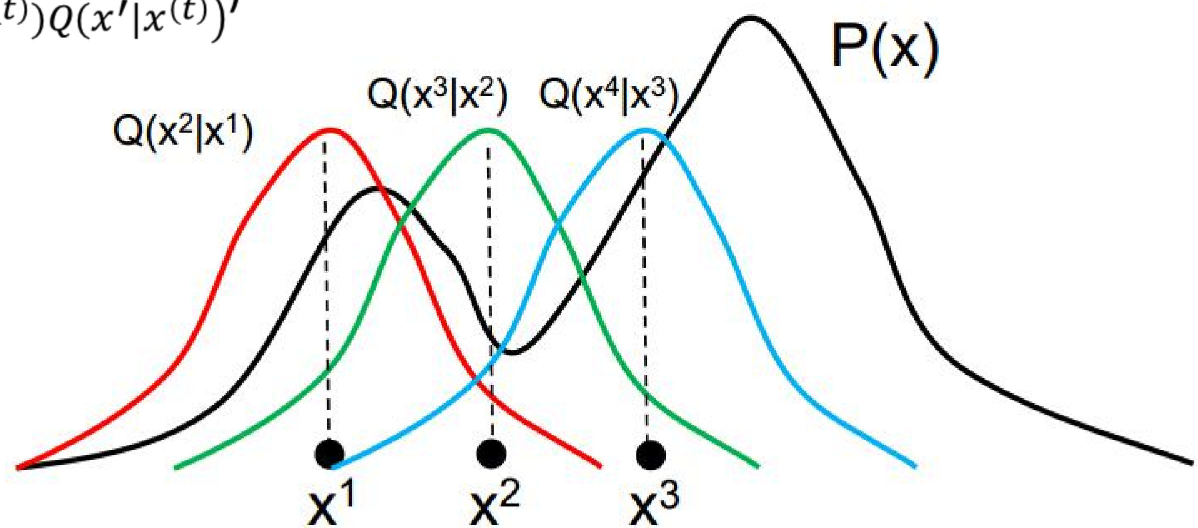
4. Repeat steps 2 and 3 until the samples "converge".



- Note that because P is in both the numerator and denominator, we can use the unnormalized P' and there is no need to find the partition function α.

- The acceptance probability $A(x'|x^{(t)})$ is like a ratio of importance sampling weights. $P(x')/Q(x'|x^{(t)})$ is the importance weight for x', $P(x^{(t)})/Q(x^{(t)}|x')$ is the importance weight for $x^{(t)}$. So it just like that We divide the importance weight for x' by that of x.

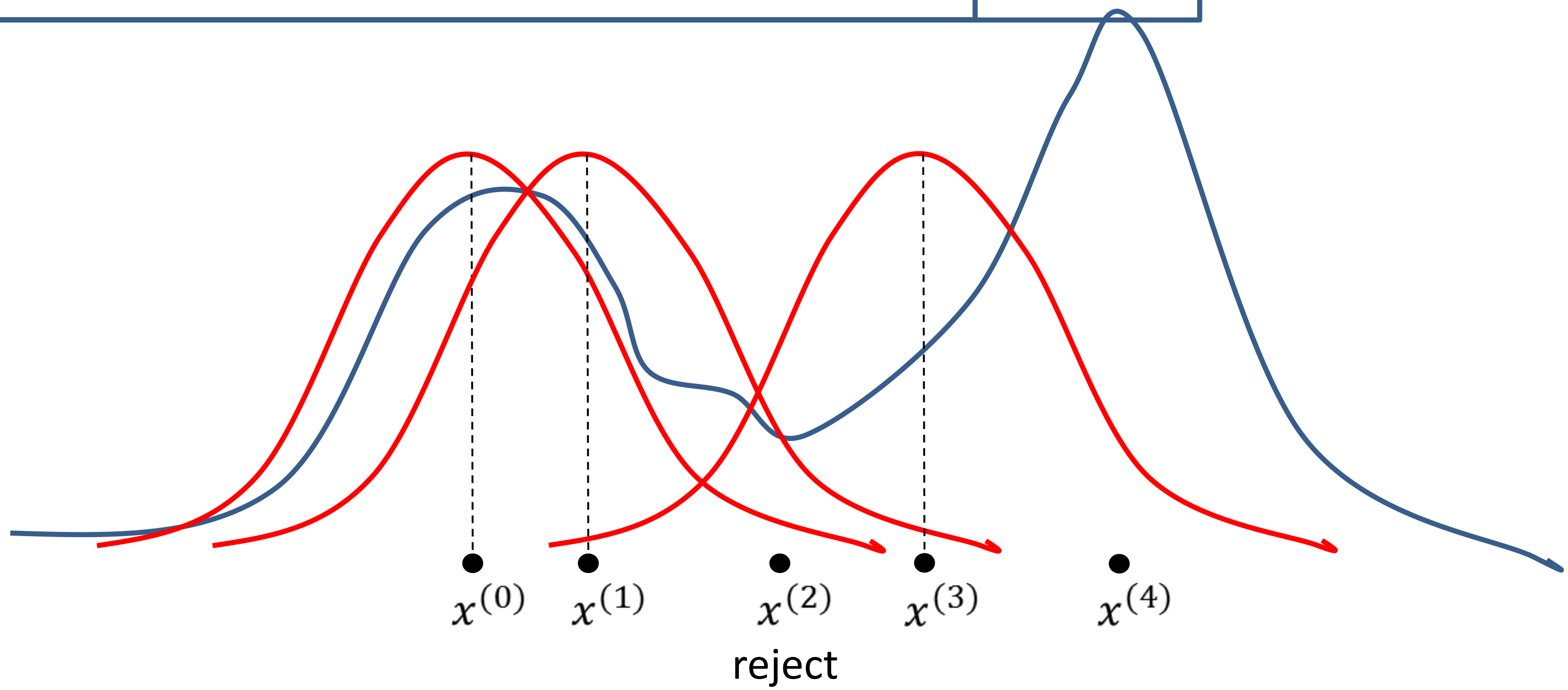
$$A(x'|x^{(t)}) = \min(1, \frac{P(x')Q(x^{(t)}|x')}{P(x^{(t)})Q(x'|x^{(t)})})$$



- An example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

We reject because $P(x^{(2)})/Q(x^{(2)}|x^{(1)}) < 1$ and $P(x^{(1)})/Q(x^{(1)}|x^{(2)}) > 1$, hence $A(x^{(2)}|x^{(1)})$ is close to zero

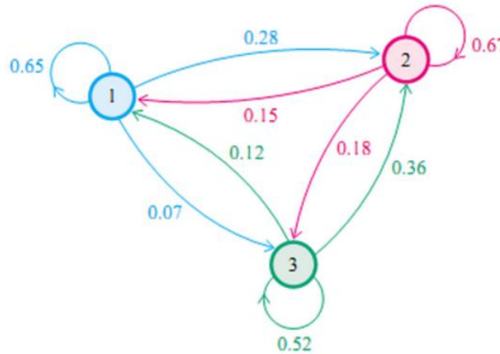
The adaptive proposal $Q(x'|x)$ allows us to sample both modes of $P(x)$!



- MC(Markov Chain) concepts:
  - A Markov Chain is a sequence of random variables $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ with the Markov Property:

$$P(x^{(n)} = x | x^{(1)}, x^{(2)}, \ldots, x^{(n-1)}) = P(x^{(n)} = x | x^{(n-1)})$$

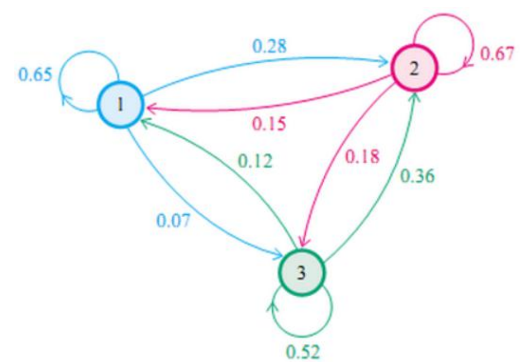
  - $x^{(i)}$ is the i-th sample of all variables in a graphical model.
  - $x^{(i)}$ represents the entire state of the graphical model at time i.
  - $P(x^{(n)} = x | x^{(n-1)})$ is known as the transition kernel.
  - The next state depends only on the preceding state.



MCMC

- We study homogeneous Markov Chains, in which the transition kernel $P(x^{(n)} = x | x^{(n-1)})$ is fixed with time:
  - For convenience, we call the kernel $T(x'|x)$, where x is the previous state and x' is the next state.

- When dealing with MCs, we don't think of the system as being in one state, but as having a distribution over states.
  - Probability distributions over states: $\pi^{(t)}(x)$ is a distribution over the state of the model, at time t.

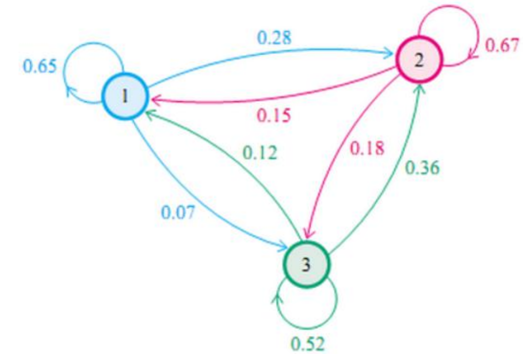


- Transitions: recall that states transition from $x^{(t)}$ to $x^{(t+1)}$ according to the transition kernel $T(x'|x)$. We can also transition entire distributions:

$$\Pi^{(t+1)}(x') = \sum_x \Pi^{(t)}(x)\, T(x'|x) \qquad \text{for all } x'$$

- Stationary distributions: is stationary if it does not change under the transition kernel:

$$\Pi(x') = \sum_x \Pi(x) T(x'|x) \qquad \text{for all } x'$$



MCMC

- Stationary distributions are of great importance in MCMC. To understand them, we need to define some notions:

  - Irreducible(不可约): an MC is irreducible if you can get from any state x to any other state x' with probability > 0 in a finite number of steps.

  - Aperiodic(非周期): an MC is aperiodic if you can return to any state x at any time.

  - Ergodic(遍历): an MC is ergodic if it is irreducible and aperiodic.

- Ergodicity is important: it implies you can reach the stationary distribution $\pi_{st}(x)$, no matter the initial distribution $\pi^{(0)}(x)$.

- Reversible(可逆)/Detailed balance(细致平稳): an MC is reversible if there exists a distribution such that the detailed balance condition is satisfied:

$$\Pi(x')T(x|x') = \Pi(x)T(x'|x)$$

- Reversibility guarantees to have a Stationary distribution:

$$\Pi(x')T(x|x') = \Pi(x)T(x'|x)$$

$$\sum_x \Pi(x')T(x|x') = \sum_x \Pi(x)T(x'|x)$$

$$\Pi(x')\sum_x T(x|x') = \sum_x \Pi(x)T(x'|x)$$

$$\Pi(x') = \sum_x \Pi(x)T(x'|x)$$

- Back to MH algorithm

  - The proposal $Q(x'|x)$ keeps changing with the value of x; how do we know the samples will eventually come from $P(x)$?

    - Recall that we draw a sample x' according to $Q(x'|x)$, and then accept/reject according to $A(x'|x)$.

  Thus here the transition kernel is

$$T(x'|x) = Q(x'|x)A(x'|x)$$

  - We can prove that MH algorithm satisfies detailed balance:

    - Recall that $A(x'|x) = \min(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)})$

    - This implies that:

      if $A(x'|x) \leq 1$, then $\frac{P(x)Q(x'|x)}{P(x')Q(x'|x)} \geq 1$ and thus $A(x'|x) = 1$

- Now suppose $A(x'|x) < 1$ and $A(x|x') = 1$, we have:

$$A(x'|x) = \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}$$

$$P(x)Q(x'|x)A(x'|x) = P(x')Q(x|x') \cdot 1$$

$$\boxed{P(x)Q(x'|x)}A(x'|x) = \boxed{P(x')Q(x|x')}A(x|x')$$

$$P(x)T(x'|x) = P(x')T(x|x')$$

The structure of $A(x'|x)$ is derived from this line!

- The last line is the detailed balance condition.
- Thus, the MH algorithm eventually converges to the target distribution $P(x)$!

Gibbs Sampling

- In the high dimension level, the acceptance rate in MH is still not high enough, so can we find a transition matrix to directly make the acceptance rate $A(x'|x) = 1$?

- Gibbs sampling is a special case of the MH method where the proposal distributions are tractable conditional distributions on $P(x)$, which can achieve the above goal.

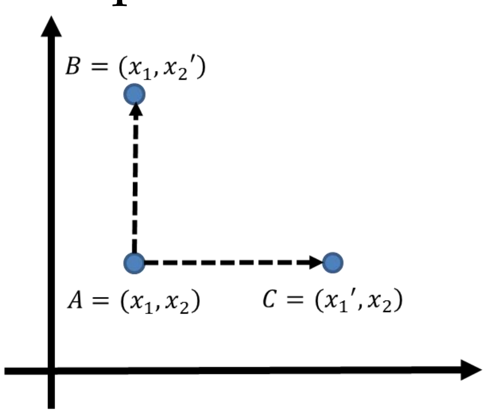
  - The idea is here:

1. Suppose in a two dimensional space, here are three points:

$A = (x_1, x_2)$

$B = (x_1, x_2')$

$C = (x_1', x_2)$



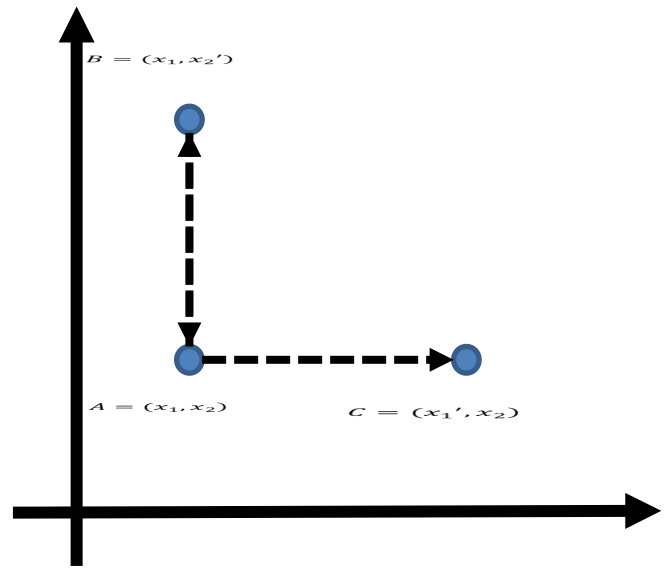
Gibbs Sampling

2. We find that:
   - For points A and B

$$P(x_1, x_2)P(x_2'|x_1) = \boxed{P(x_1)P(x_2|x_1)P(x_2'|x_1)}$$

$$\|$$

$$P(x_1, x_2')P(x_2|x_1) = \boxed{P(x_1)P(x_2'|x_1)P(x_2|x_1)}$$



$\Rightarrow$ $$P(x_1, x_2)P(x_2'|x_1) = P(x_1, x_2')P(x_2|x_1)$$

$\Rightarrow$ $$\boxed{P(A)P(x_2'|x_1) = P(B)P(x_2|x_1)}$$

Similarly $$P(A)P(x_1'|x_2) = P(C)P(x_1|x_2)$$

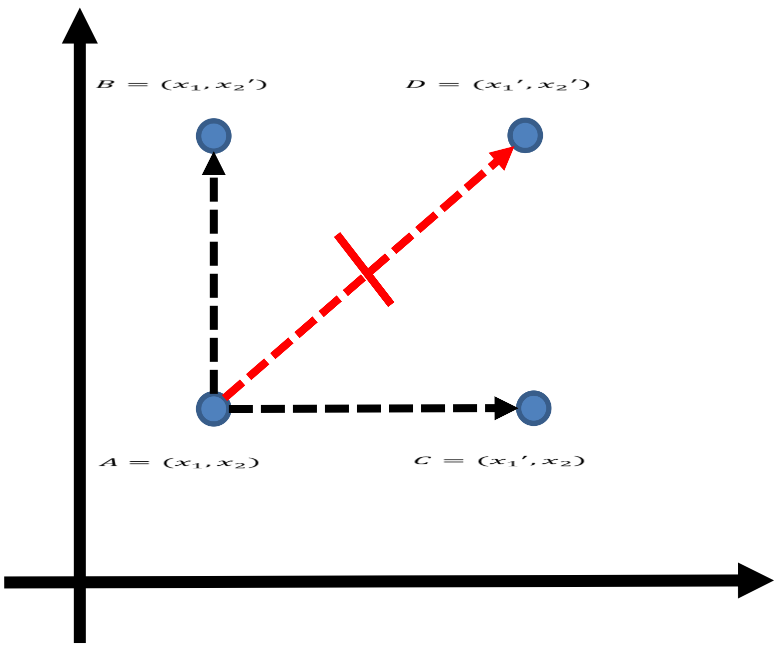
Gibbs Sampling

3. Thus we can build the transition matrix:

$$
\begin{cases}
Q(A \to B) = P(x_2 | x_1) & \text{if } x_1(A) = x_1(B) = x_1 \\
Q(A \to C) = P(x_1 | x_2) & \text{if } x_2(A) = x_2(C) = x_2 \\
Q(A \to D) = 0 & \text{otherwise}
\end{cases}
$$

- It's easy to see that this transition matrix satisfies the detailed balance condition. So it will lead to the stationary distribution.

- Gibbs sampling 是一个“多重转移模型”(Multiple Transition Models), 每一个核就是延一个坐标轴的转移，单个核不足以保证马尔科夫链的遍历性，但多核则可以使其收敛于稳定分布。每次我们随机或轮转选择其中一个核。



Gibbs Sampling

- The pseudocode:

Algorithm 8 n维Gibbs Sampling 算法

1: 随机初始化$\{x_i : i = 1, \cdots, n\}$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $x_1^{(t+1)} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \cdots, x_n^{(t)})$

2. $x_2^{(t+1)} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \cdots, x_n^{(t)})$

3. $\cdots$

4. $x_j^{(t+1)} \sim p(x_j | x_1^{(t+1)}, \cdots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \cdots, x_n^{(t)})$

5. $\cdots$

6. $x_n^{(t+1)} \sim p(x_n | x_1^{(t+1)}, x_2^{t}, \cdots, x_{n-1}^{(t+1)})$

Gibbs Sampling

- We will show the Gibbs sampling is a particular case of MH method, whose acceptance rate is 1.

  - Define x_i to be the i-th element of the feature vector x and x_{-i} to be all other elements. Gibbs sampling let

$$Q(x'|x) = Q(x_i', x_{-i}|x_i, x_{-i}) = P(x_i'|x_{-i})$$

  - Then

$$A(x_i', x_{-i}|x_i, x_{-i}) = \min(1, \frac{P(x_i', x_{-i})Q(x_i, x_{-i}|x_i', x_{-i})}{P(x_i, x_{-i})Q(x_i', x_{-i}|x_i, x_{-i})})$$

$$= \min(1, \frac{P(x_i', x_{-i})P(x_i|x_{-i})}{P(x_i, x_{-i})P(x_i'|x_{-i})})$$

$$= \min(1, \frac{P(x_i'|x_{-i})P(x_{-i})P(x_i|x_{-i})}{P(x_i|x_{-i})P(x_{-i})P(x_i'|x_{-i})})$$

$$= \min(1,1) \quad = 1$$

Problems of MCMC

- It can be hard to move from one high probability space to another across a low probability space.

- The samples are not independent with each other truly, especially in Gibbs sampling. How to determine how two samples are “far enough” to be considered independent draws.

- Although MH algorithm will converge to the true distribution, with certain exceptions, there are no guarantees to when. In fact, it’s an art to decide when to stop the algorithm.

- …….

More Advanced Topics of Particle-Based Methods

- The collapsed particles(坍塌的粒子).
- Deterministic Search Methods(确定性搜索方法).

Reference

- Daphne Koller and Nir Friedman. Probabilistic Graphical Models: Principles and Techniques. Ch.12.
- Christopher Bishop, et al. Pattern Recognition And Machine Learning. Ch.11.
- Eric Xin. Probabilistic Graphical Models. Lecture 16, 17, 18.
- rickjin. LDA-math-MCMC and Gibbs Sampling. http://cos.name/2013/01/lda-math-mcmc-and-gibbs-sampling/

Thanks