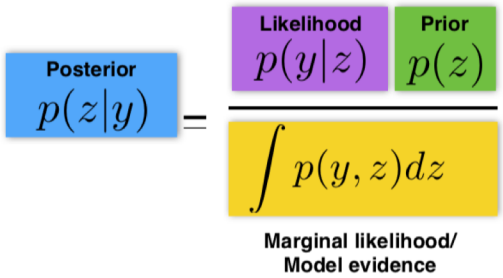
Approximate Inference
Monte Carlo Methods

Kayhan Batmanghelich

Inferential Problems



Most inference problems will be one of:

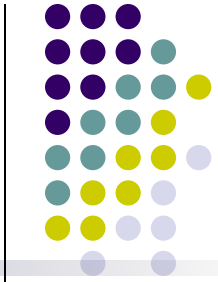
Marginalisation $p(y) = \int p(y, \theta) d\theta$

Expectation $\mathbb{E}[f(y)|x] = \int f(y)p(y|x)dy$

Prediction $p(y_{t+1}) = \int p(y_{t+1}|y_t)p(y_t)dy_t$

Slides Credit: Shakir Mohamed (DeepMind)

Approaches to inference



- Exact inference algorithms
  - The elimination algorithm
  - Message-passing algorithm (sum-product, belief propagation)
  - The junction tree algorithms
- Approximate inference techniques
  - Variational algorithms
    - Loopy belief propagation
    - Mean field approximation
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods

Properties of Monte Carlo

Estimator: $\int f(x)P(x)\,\mathrm{d}x \approx \hat{f} \equiv \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$

Estimator is unbiased:

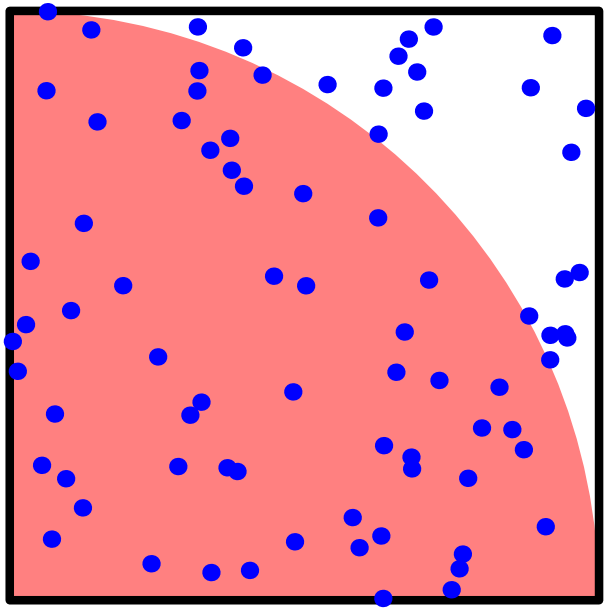
$$\mathbb{E}_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{P(x)}\left[f(x)\right] = \mathbb{E}_{P(x)}\left[f(x)\right]$$

Variance shrinks $\propto 1/S$**:**

$$\mathrm{var}_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{var}_{P(x)}\left[f(x)\right] = \mathrm{var}_{P(x)}\left[f(x)\right]/S$$

"Error bars" shrink like $\sqrt{S}$

A dumb approximation of π



$$P(x,y) = \begin{cases} 1 & 0<x<1 \text{ and } 0<y<1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}\left((x^2+y^2)<1\right) P(x,y)\, \mathrm{d}x\, \mathrm{d}y$$

```
octave:1> S=12; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.3333
octave:2> S=1e7; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.1418
```

Aside: don't always sample!

"Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse."

— Alan Sokal, 1996

Example: numerical solutions to (nice) 1D integrals are fast

```
octave:1> 4 * quadl(@(x) sqrt(1-x.^2), 0, 1, tolerance)
```

Gives π to 6 dp's in 108 evaluations, machine precision in 2598.

(NB Matlab's `quadl` fails at zero tolerance)

Sampling from distributions

How to convert samples from a Uniform[0,1] generator:

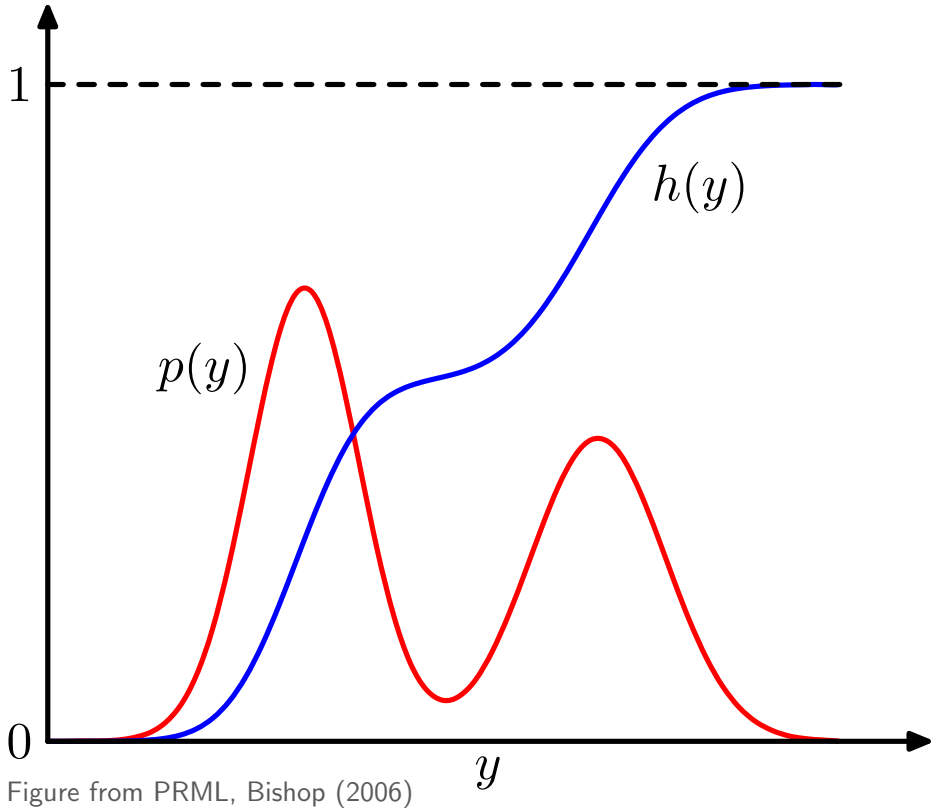


Figure from PRML, Bishop (2006)

$$h(y) = \int_{-\infty}^{y} p(y')\,\mathrm{d}y'$$

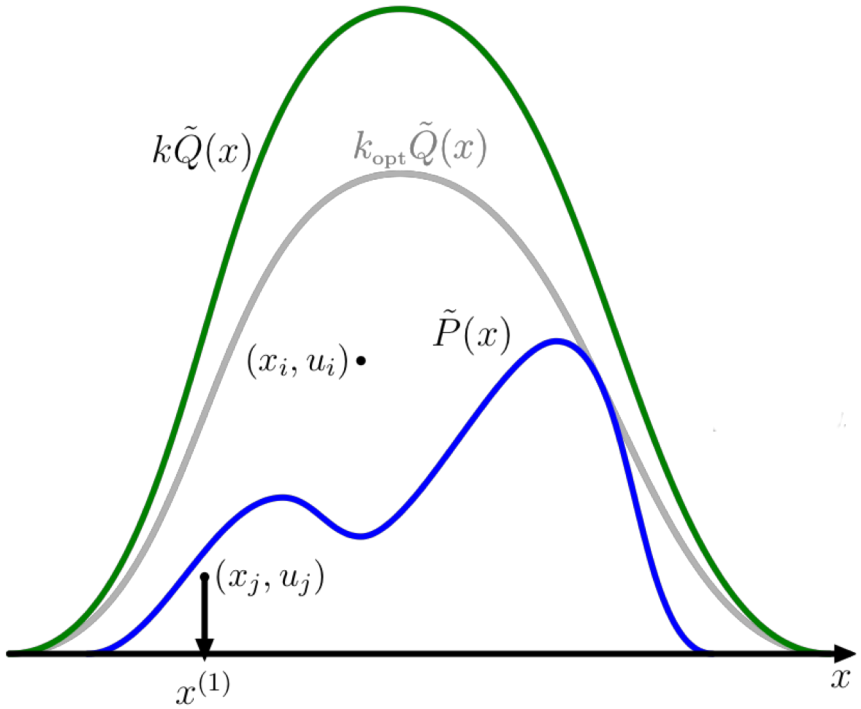
Draw mass to left of point:

$u \sim$ Uniform[0,1]

Sample, $y(u) = h^{-1}(u)$

Although we can't always compute and invert $h(y)$

Rejection Sampling



Steps:

- Find Q(x) that is easy to sample from.
- Find k such that k such that:

$$\frac{\tilde{P}(x)}{kQ(x)} < 1$$

- Sample auxiliary variable y

$$\mathbb{P}(y = 1|x) = \frac{\tilde{P}(x)}{kQ(x)}$$

accept the sample with probability P(y=1|x)

Claim: Accepted samples have a probability of P(x).
Does it matter how to choose k ?

Pitfalls of Rejection Sampling

Rejection & importance sampling scale badly with dimensionality

Example:

$$P(x) = \mathcal{N}(0, \mathbb{I}), \quad Q(x) = \mathcal{N}(0, \sigma^2\mathbb{I})$$

the densities are fully factorizable in this example:

$$p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i) \qquad q(\mathbf{x}) = \prod_{i=1}^{D} q(x_i)$$

The acceptance rate is:

$$q(y = 1|\mathbf{x}) = \prod_{i=1}^{D} \frac{p^*(x_i)}{M_i q(x_i)} = \prod_{i=1}^{D} q(y = 1|x_i) = O\left(\gamma^D\right)$$

Importance sampling

Computing $\tilde{P}(x)$ and $\tilde{Q}(x)$, then *throwing* x *away* seems wasteful
Instead rewrite the integral as an expectation under Q:

$$\int f(x)P(x)\,\mathrm{d}x = \int f(x)\frac{P(x)}{Q(x)}Q(x)\,\mathrm{d}x, \qquad (Q(x) > 0 \text{ if } P(x) > 0)$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\frac{P(x^{(s)})}{Q(x^{(s)})}, \quad x^{(s)} \sim Q(x)$$

We switched the sampling from P(x) which is hard to sampling from Q(x).

Wait!! We still need to have $\frac{P(x^S)}{Q(x^S)}$.

Importance Sampling (2)

Previous slide assumed we could evaluate $P(x) = \tilde{P}(x)/\mathcal{Z}_P$

$$\int_x f(x)p(x) = \frac{\int_x f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)}{\int_x \frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)}$$

Let $x^1, \cdots, x^L$ be samples from q(x).

$$\int_x f(x)p(x) \approx \frac{\sum_l f(x^l)\frac{\tilde{p}(x^l)}{\tilde{q}(x^l)}}{\sum_l \frac{\tilde{p}(x^l)}{\tilde{q}(x^l)}} = \sum_{l=1}^{L} f(x^l)w_l$$

This estimator is **consistent** but **biased**

What is the implication?

Exercise: Prove that $\frac{Z_p}{Z_Q} \approx \frac{1}{L}\sum_L \tilde{w}_l$

Pitfalls of Importance Sampling

Naïve importance sampling does not scale well with dimensionality

- The proposal distribution (q(x)) is a good one when p=q.
- In other words, weighs are uniform (w=1/L).
- Let's study variability of the unnormalized weights

$$u_i = p(\mathbf{x}^i)/q(\mathbf{x}^i)$$

$$\left\langle (u_i - u_j)^2 \right\rangle = \left\langle u_i^2 \right\rangle + \left\langle u_j^2 \right\rangle - 2 \left\langle u_i \right\rangle \left\langle u_j \right\rangle$$

Example: Fully factorizable p(x) and q(x) :

$$\left\langle (u_i - u_j)^2 \right\rangle = 2 \left(\left\langle \frac{p(x)}{q(x)} \right\rangle_{p(x)}^{D} - 1 \right)$$

Prove it!

$$\left\langle \frac{p(x)}{q(x)} \right\rangle_{p(x)} > 1$$

Pitfalls of Importance Sampling

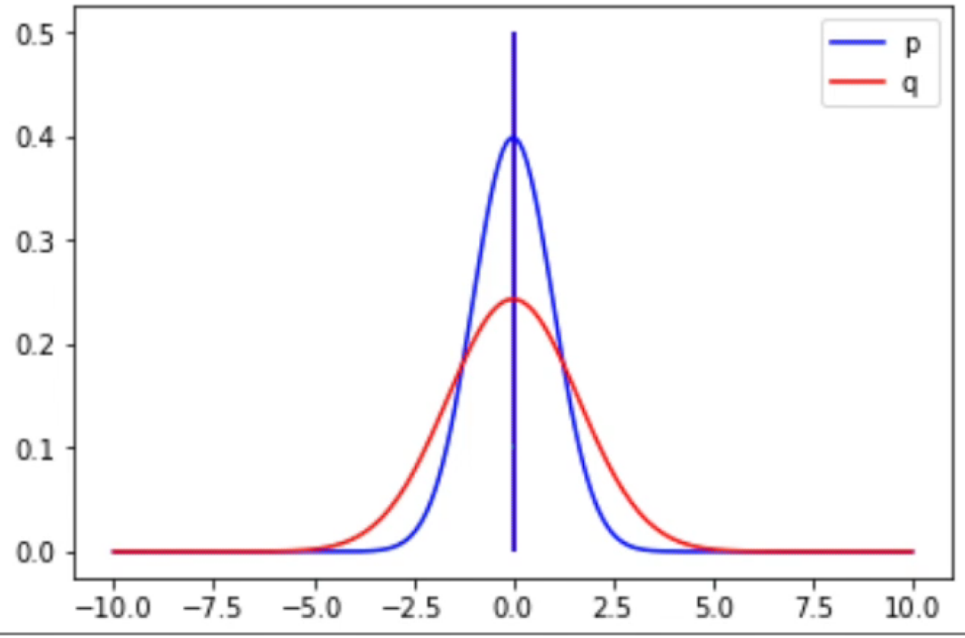
```
interact(myPlot, log_s=(-3,5,0.01),mu=(-8,8,0.5))
```

log_s 0.99

mu 0.00

```
-0.000294354607243
```

Out[11]: <function __main__.myPlot>



Pitfalls of Importance Sampling

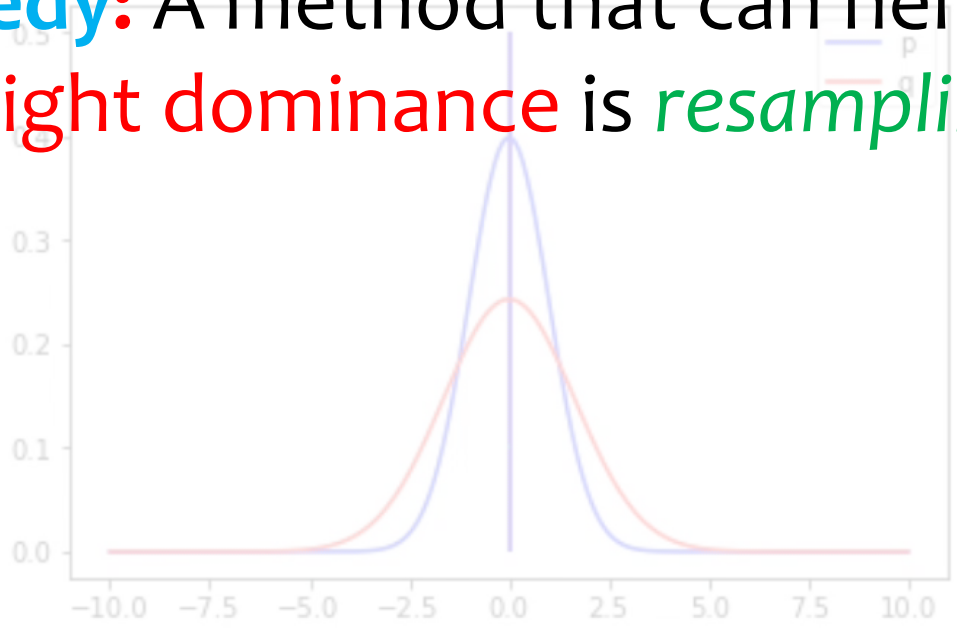
```
interact(myPlot, log_s=(-3,5,0.01),mu=(-8,8,0.5))
```

log_s 0.99

mu 0.00

```
-0.000294354607243
Out[11]: <function __main__.myPlot>
```

A Remedy: A method that can help address this weight dominance is *resampling*.



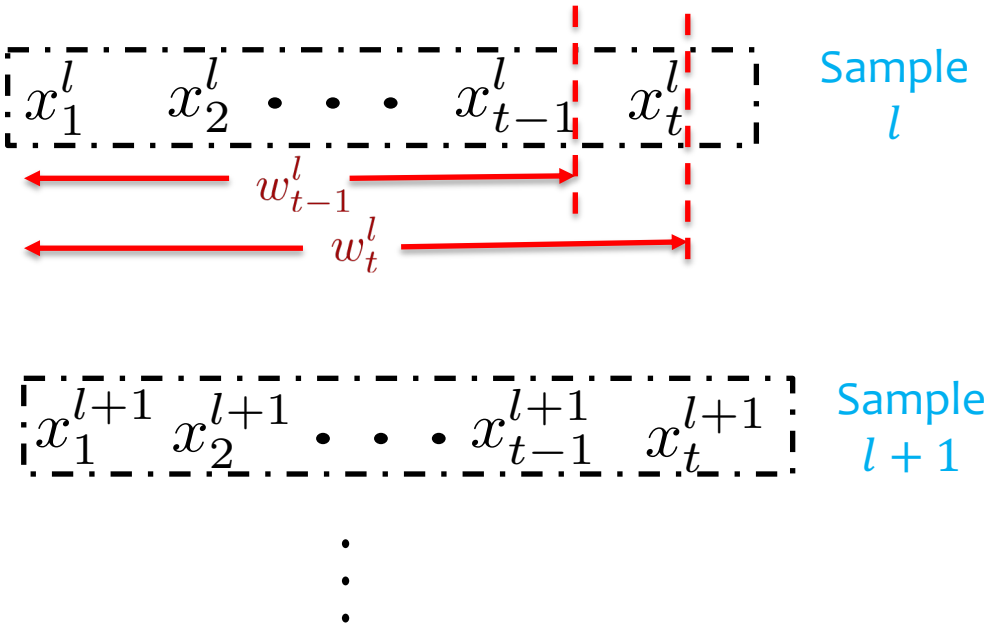
How to use structure in high dim?

Apply the importance sampling to a temporal distribution $p(x_{1:t})$

General idea, change the proposal in each step: $q(x_t|x_{1:t})$

$$\tilde{w}_t^l = \frac{p^*(x_{1:t}^l)}{q(x_{1:t}^l)}$$

$$= \frac{p^*(x_t^l|x_{1:t-1}^l)}{q(x_t^l|x_{1:t-1}^l)} \frac{p^*(x_{1:t-1}^l)}{q(x_{1:t-1}^l)}$$



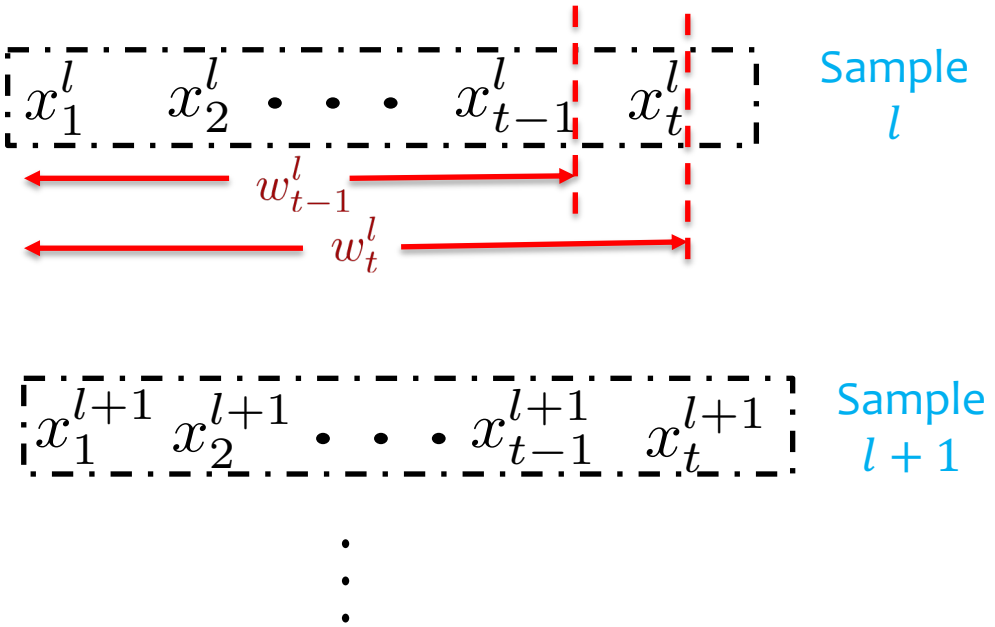
How to use structure in high dim?

Apply the importance sampling to a temporal distribution $p(x_{1:t})$

General idea, change the proposal in each step: $q(x_t|x_{1:t})$

$$\tilde{w}_t^l = \frac{p^*(x_{1:t}^l)}{q(x_{1:t}^l)}$$

$$= \frac{p^*(x_t^l|x_{1:t-1}^l)}{q(x_t^l|x_{1:t-1}^l)} \frac{p^*(x_{1:t-1}^l)}{q(x_{1:t-1}^l)}$$



The recursion rule:

$$\tilde{w}_t^l = \tilde{w}_{t-1}^l \alpha_t^l, \qquad t > 1$$

$$\alpha_t^l \equiv \frac{p^*(x_t^l|x_{1:t-1}^l)}{q(x_t^l|x_{1:t-1}^l)}$$

Sketch of Particle Filters

Apply the importance sampling to a temporal distribution $p(x_{1:t})$

General idea, change the proposal in each step: $q(x_t|x_{1:t})$

The recursion rule:

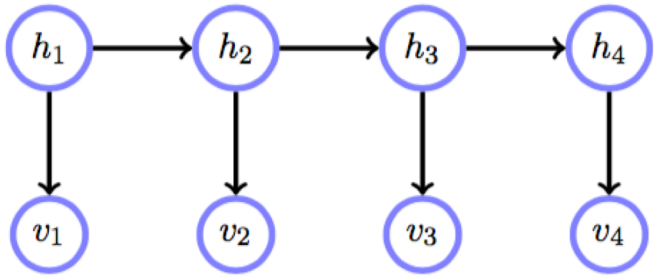
$$\tilde{w}_t^l = \tilde{w}_{t-1}^l \alpha_t^l, \qquad t > 1$$

$$\alpha_t^l \equiv \frac{p^*(x_t^l|x_{1:t-1}^l)}{q(x_t^l|x_{1:t-1}^l)}$$

$$\alpha_t^l \equiv \frac{p(v_t|h_t^l)p(h_t^l|h_{t-1}^l)}{q(h_t^l|h_{1:t-1}^l)}$$

$$q(h_t|h_{1:t-1}) = p(h_t|h_{t-1})$$

$$\tilde{w}_t^l = \tilde{w}_{t-1}^l p(v_t|h_t^l)$$



Sketch of Particle Filters

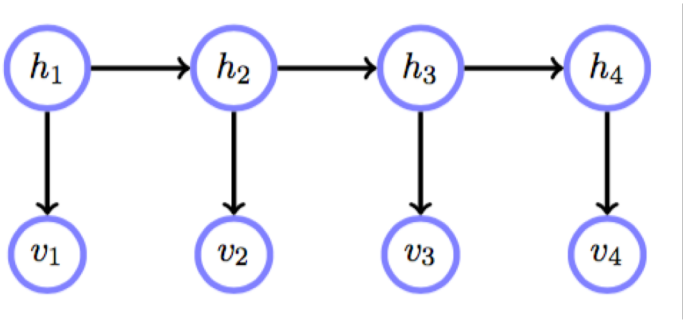
Apply the importance sampling to a temporal distribution $p(x_{1:t})$

General idea, change the proposal in each step: $q(x_t|x_{1:t})$

The recursion rule:

$$q(h_t|h_{1:t-1}) = p(h_t|h_{t-1})$$

$$\tilde{w}_t^l = \tilde{w}_{t-1}^l p(v_t|h_t^l)$$



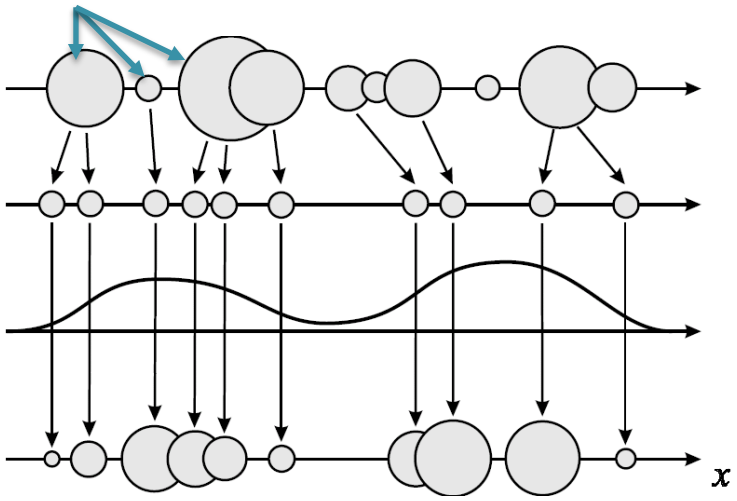
Forward message:

$$\rho(h_t) \propto p(h_t|v_{1:t})$$

$$\rho(h_t) \propto p(v_t|h_t) \int_{h_{t-1}} p(h_t|h_{t-1}) \rho(h_{t-1})$$

$$\rho(h_{t-1}) \approx \sum_{l=1}^{L} w_{t-1}^l \delta\left(h_{t-1}, h_{t-1}^l\right)$$

$$\rho(h_t) \approx \frac{1}{Z} p(v_t|h_t) \sum_{l=1}^{L} p(h_t|h_{t-1}^l) w_{t-1}^l$$



Summary so far

General ideas for the sampling approaches

- Proposal distribution (q(x)): Use another distribution to sample from.
  - Change the proposal distribution with the iterations.

- Introduce an auxiliary variable to decide keeping a sample or not.
  - Why should we discard samples?

- Sampling from high-dimension is difficult.
  - Let's incorporate the graphical model into our sampling strategy.

- Can we use the gradient of the p(x)?

Summary so far

General ideas for the sampling approaches

- Proposal distribution (q(x)): Use another distribution to sample from.
  - Change the proposal distribution with the iterations.

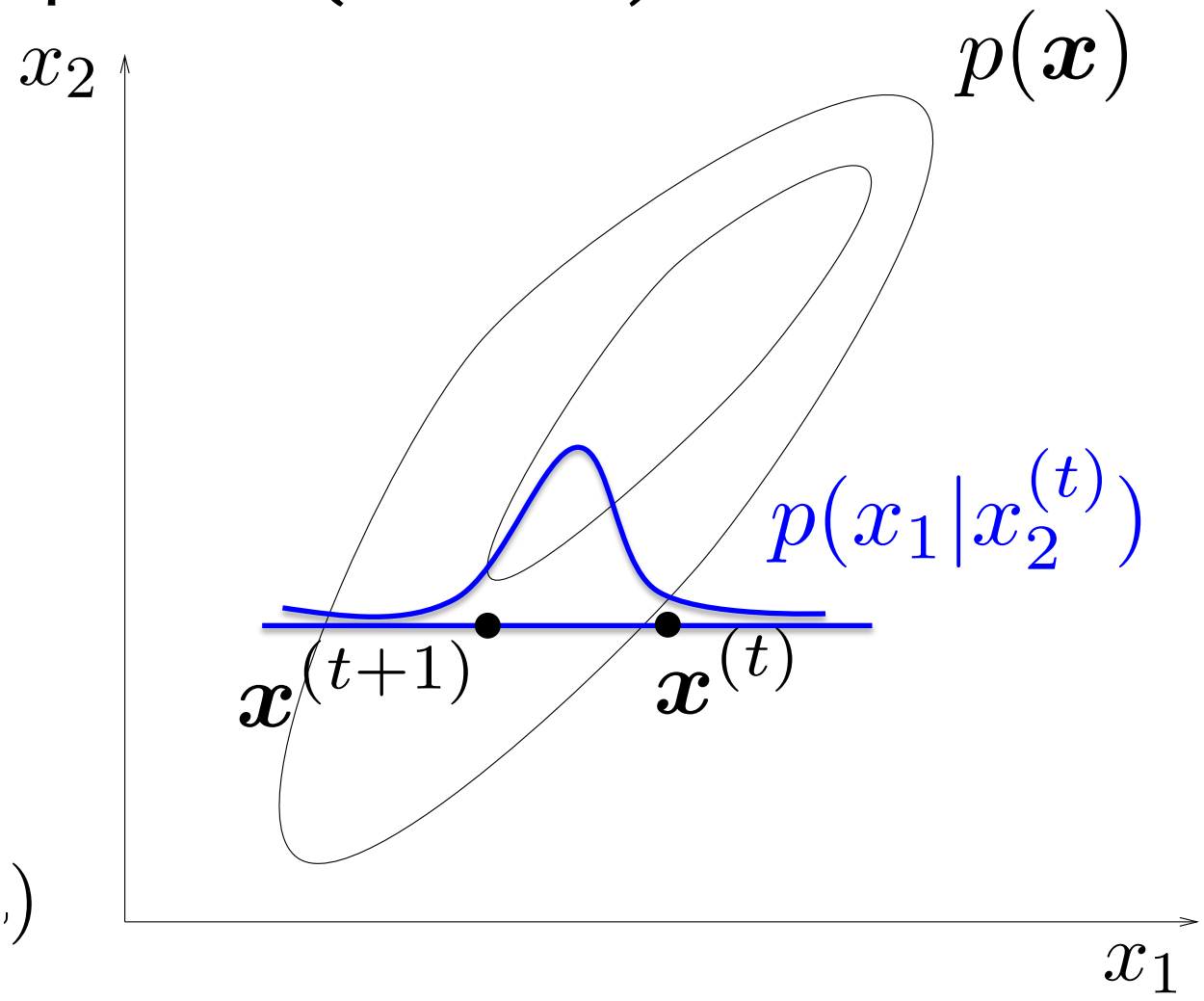
- Introduce an auxiliary variable to decide keeping a sample or not.
  - Why should we discard samples?

- Sampling from high-dimension is difficult.
  - Let's incorporate the graphical model into our sampling strategy.

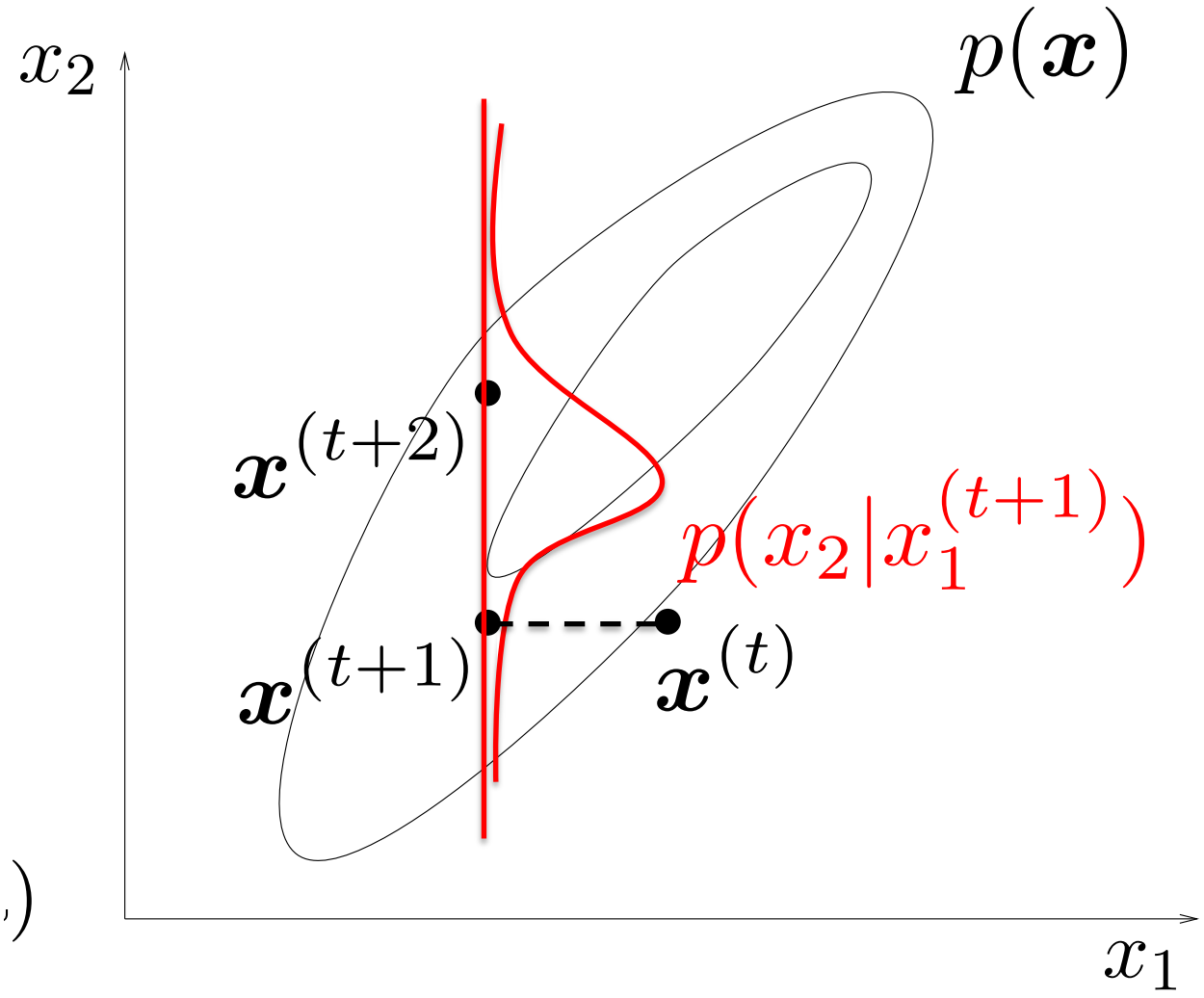
- Can we use the gradient of the p(x)?

Gibbs Sampling

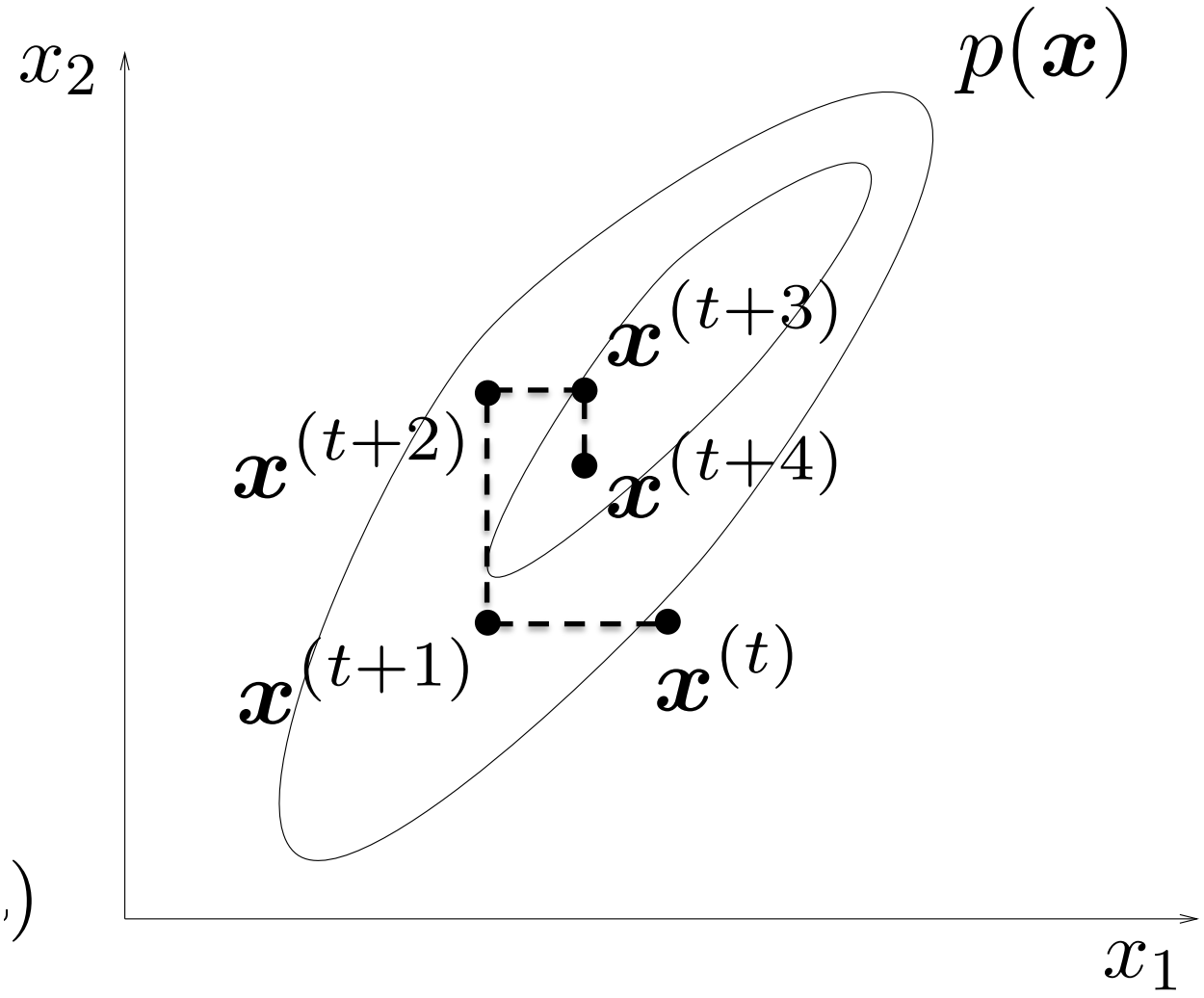
- Sample one (block of) variable at the time



Gibbs Sampling



Gibbs Sampling



Gibbs Sampling

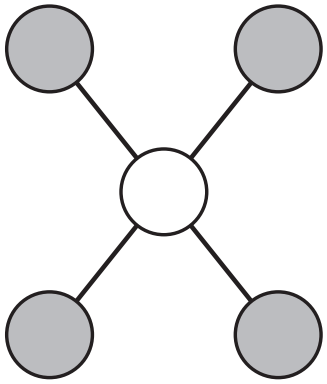
Link:
https://www.youtube.com/watch?v=AEwY6QXWoUg
https://www.youtube.com/watch?v=ZaKwpVgmKTY

Figure from Jordan Ch. 21

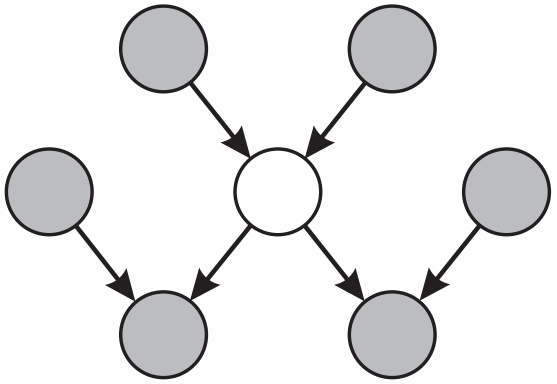
Ingredients for Gibb Recipe

Full conditionals only need to condition on the Markov Blanket

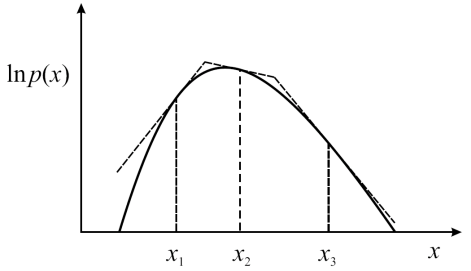
MRF



Bayes Net



- Must be "easy" to sample from conditionals
- Many conditionals are log-concave and are amenable to adaptive rejection sampling



Gibbs Sampling

- Sample one (block of) variable at the time

$$p(x) = p(x_i|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)p(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$$

$$p(x_i|x_{\setminus i}) = \frac{1}{Z}p(x_i|\mathrm{pa}\,(x_i)) \prod_{j\in \mathrm{ch}(i)} p(x_j|\mathrm{pa}\,(x_j))$$

Markov Blanket

Easy to compute

- The proposal distribution:

$$q(x^{l+1}|x^l,i) = p(x_i^{l+1}|x_{\setminus i}^l)\prod_{j\neq i}\delta\left(x_j^{l+1},x_j^l\right)$$

Make sure other variables do not change

$$q(x^{l+1}|x^l) = \sum_i q(x^{l+1}|x^l,i)q(i),$$

Choose one of the variables randomly with probability q(i)

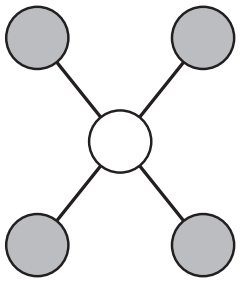
Figure from Jordan Ch. 21

Again….

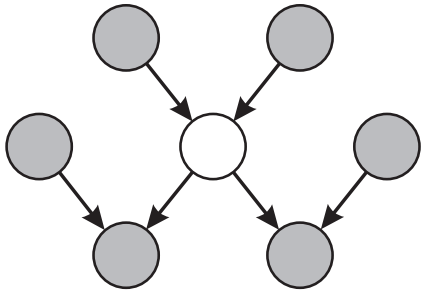
$$p(x_i|x_{\setminus i}) = \frac{1}{Z} p(x_i|\text{pa}(x_i)) \prod_{j \in \text{ch}(i)} p(x_j|\text{pa}(x_j))$$

Full conditionals only need to condition on the Markov Blanket

MRF

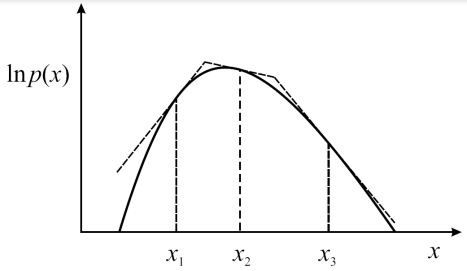


Bayes Net



$$p(x_i|x_{\setminus i}) = \frac{1}{Z} p(x_i|\text{pa}(x_i)) \prod_{j \in \text{ch}(i)} p(x_j|\text{pa}(x_j))$$

- Must be "easy" to sample from conditionals
- Many conditionals are log-concave and are amenable to adaptive rejection sampling

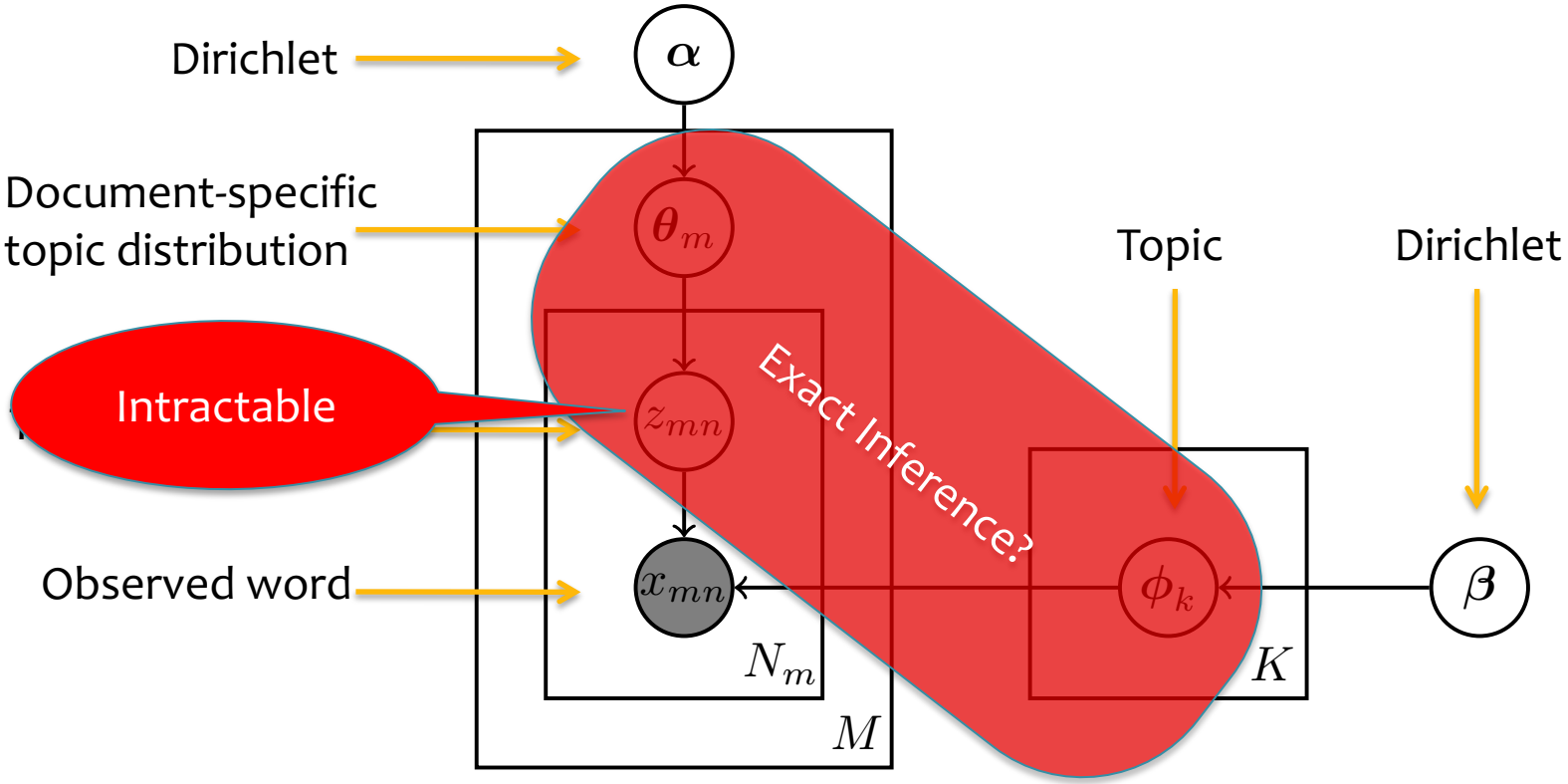


Whiteboard

- Gibbs Sampling as M-H

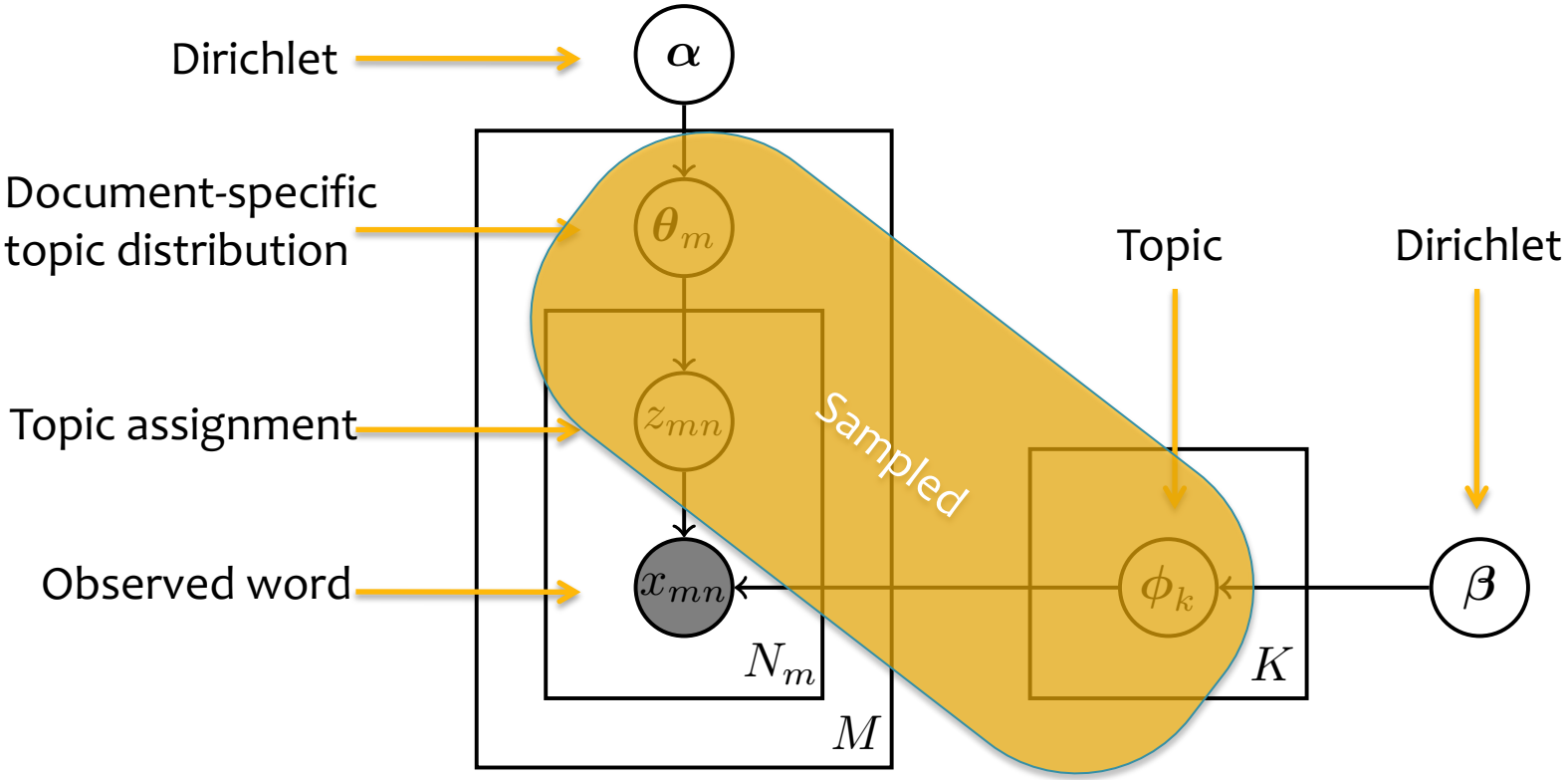
LDA Inference

- Bayesian Approach



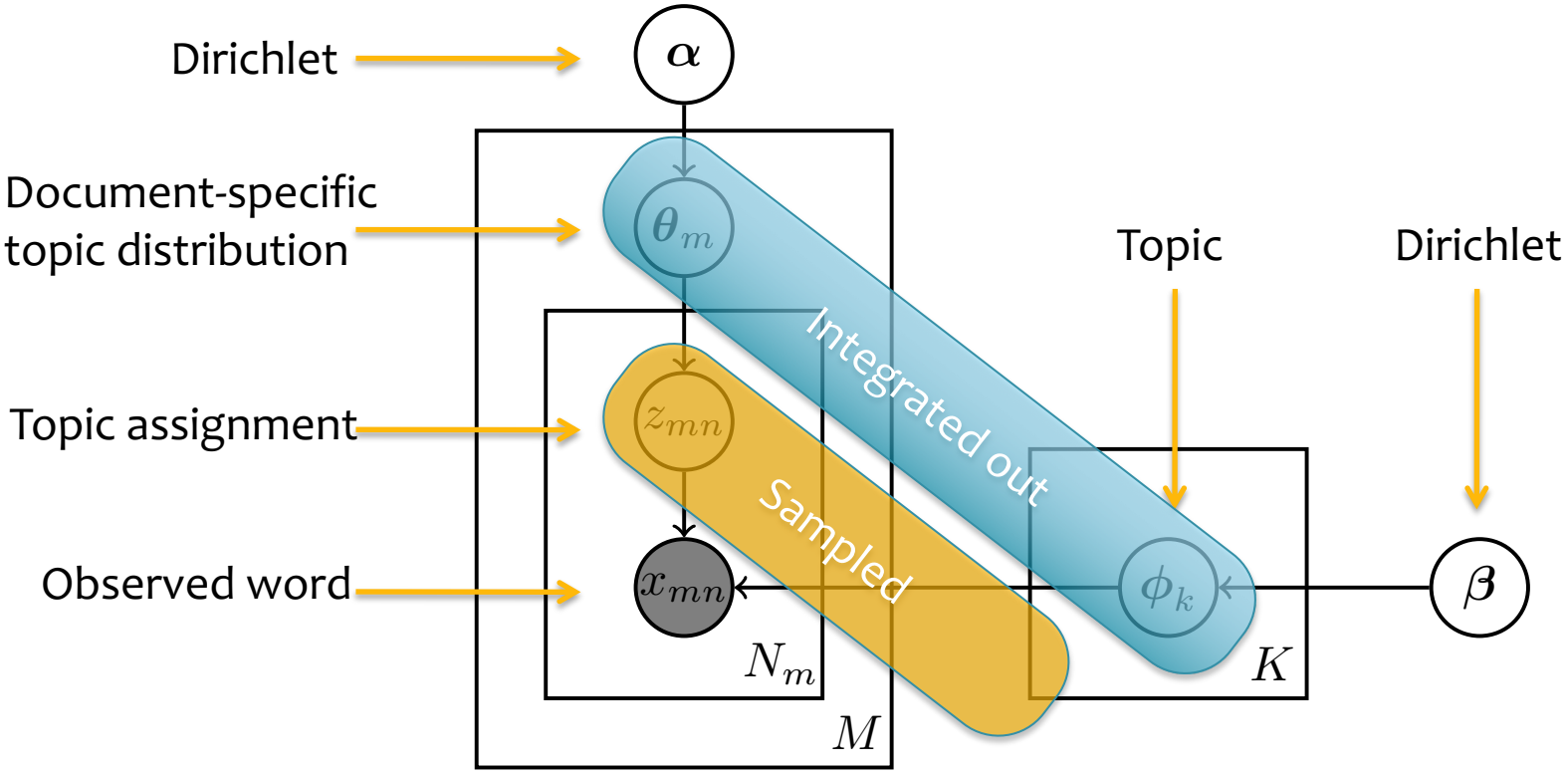
LDA Inference

- Explicit Gibbs Sampler



LDA Inference

- Collapsed Gibbs Sampler



Sampling

Goal:

- Draw samples from the posterior $p(Z|X, \alpha, \beta)$
- Integrate out topics ϕ and document-specific distribution over topics θ

Algorithm:

- While not done…
  - For each document, m:
    - For each word, n:
      - » Resample a single topic assignment using the full conditionals for z_{mn}

Sampling

- What queries can we answer with samples of z_{mn}?
  - Mean of z_{mn}
  - Mode of z_{mn}
  - Estimate posterior over z_{mn}
  - Estimate of topics ϕ and document-specific distribution over topics θ

Gibbs Sampling for LDA

- Full conditionals

$$p(z_i = j | z_{-i}, X, \alpha, \beta) \propto \frac{n_{-i,j}^{(x_i)} + \beta}{n_{-i,j}^{(\cdot)} + T\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + K\alpha}$$

$n_{-i,j}^{(x_i)}$ the number of instances of word x_i assigned to topic j, not including current word.

$n_{-i,j}^{(\cdot)}$ total number of words assigned to topic j, not including the current one.

$n_{-i,j}^{(d_i)}$ the number of words for document d_i assigned to topic j.

$n_{-i,\cdot}^{(d_i)}$ total number of words in the document d_i not including the current one.

Link: https://people.cs.umass.edu/~wallach/courses/s11/cmpsci791ss/readings/griffiths02gibbs.pdf

Gibbs Sampling for LDA

- Sketch of the derivation of the full conditionals

$$
\begin{aligned}
p(z_i = k | Z^{-i}, X, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{p(X, Z | \boldsymbol{\alpha}, \boldsymbol{\beta})}{p(X, Z^{-i} | \boldsymbol{\alpha}, \boldsymbol{\beta})} \\
&\propto p(X, Z | \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= p(X | Z, \boldsymbol{\beta}) p(Z | \boldsymbol{\alpha}) \\
&= \int_{\Phi} p(X | Z, \Phi) p(\Phi | \boldsymbol{\beta}) \, d\Phi \int_{\Theta} p(Z | \Theta) p(\Theta | \boldsymbol{\alpha}) \, d\Theta \\
&= \left(\prod_{k=1}^{K} \frac{B(\vec{n}_k + \boldsymbol{\beta})}{B(\boldsymbol{\beta})} \right) \left(\prod_{m=1}^{M} \frac{B(\vec{n}_m + \boldsymbol{\alpha})}{B(\boldsymbol{\alpha})} \right)
\end{aligned}
$$

$$
p(z_i = j | z_{-i}, X, \alpha, \beta) \propto \frac{n_{-i,j}^{(x_i)} + \beta}{n_{-i,j}^{(\cdot)} + T\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + K\alpha}
$$

Gibbs Sampling for LDA

Algorithm

// initialisation

zero all count variables, $n_m^{(k)}, n_m, n_k^{(t)}, n_k$

for all documents $m \in [1, M]$ **do**

  for all words $n \in [1, N_m]$ in document m **do**

    sample topic index $z_{m,n}=k \sim \text{Mult}(1/K)$

    increment document–topic count: $n_m^{(k)}$ += 1

    increment document–topic sum: n_m += 1

    increment topic–term count: $n_k^{(t)}$ += 1

    increment topic–term sum: n_k += 1

Gibbs Sampling for LDA

Algorithm

// Gibbs sampling over burn-in period and sampling period

while not finished **do**

  for all documents $m \in [1, M]$ **do**

    for all words $n \in [1, N_m]$ in document m **do**

      // for the current assignment of k to a term t for word $w_{m,n}$:

      decrement counts and sums: $n_m^{(k)} -= 1; n_m -= 1; n_k^{(t)} -= 1; n_k -= 1$

      // multinomial sampling acc. to Eq. 78 (decrements from previous step):

      sample topic index $\tilde{k} \sim p(z_i | \vec{z}_{\neg i}, \vec{w})$

      // for the new assignment of $z_{m,n}$ to the term t for word $w_{m,n}$:

      increment counts and sums: $n_m^{(\tilde{k})} += 1; n_m += 1; n_{\tilde{k}}^{(t)} += 1; n_{\tilde{k}} += 1$