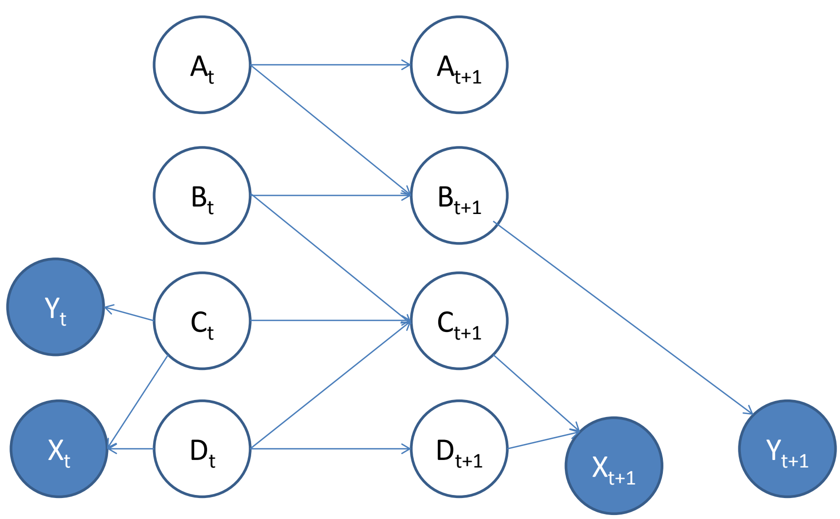
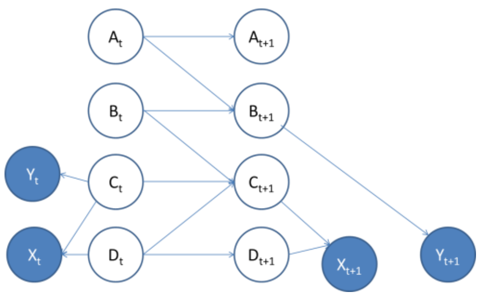
Reasoning with uncertainty IV

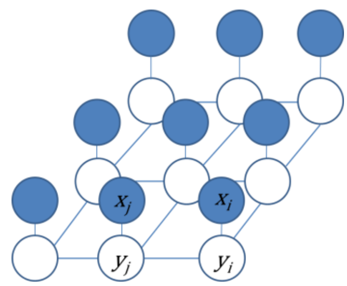


- Arbitrary connections between state and observation variables at any time t
  1. Replicate over time (unroll) → General graph, can't do exact inference directly (in general)
  2. Collapse state variables wrt observed → K^D state tables in general
- In the discrete case, DBN <=> HMM but note the complexity issue
- Alternative
  - Sampling
  - Variational, Assumed density

Approximate inference

- In general: Cannot compute $P(X_i)$ or $P(X_1,..,X_n)$ directly
- Need to use approximation
  - Sampling
  - Define tractable simpler P' and find approximation





Sampling from distribution

- Given known distribution $P(x)$ always possible to draw samples from $P(x)$
- In general (e.g., non-tree models) $P(x_1,..,x_n)$ cannot be represented explicitly → Cannot sample directly
- How to/why use samples:
  - Use distribution to compute statistics, e.g., expectations

$$E_P(f) = \int f(x)p(x) \approx \frac{1}{N}\sum_i f(x_i)$$

x_i must be independent and follow the distribution P!

Sampling

- First (simple and silly) example on a couple of Bayes nets
  - Ancestral and likelihood sampling
- General techniques
  - Rejection
  - Importance
  - MCMC
  - Gibbs
  - Sequential (particles)

Approximate Method: Sampling

- General idea:
  - It is often difficult to compute and represent exactly the probability distribution of a set of variables
  - But, it is often easy to generate examples from the distribution

P ➡

X_1 X_2...................... X_m
T T F T F F…T F
T F T T T F…T T
F T T F F F…T F
…….
F F F T F T…F T

For a large number of samples,

$P(X_1=x_1, X_2=x_2, \ldots, X_m = x_m)$

is approximately equal to:

$$\frac{\#\text{ of samples with } X_1=x_1 \text{ and } X_2=x_2 \ldots\text{and } X_m = x_m}{\text{Total }\#\text{ of samples}}$$

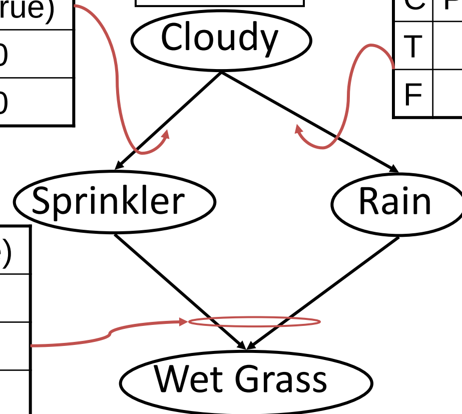
Sampling Example

- Generate a set of variable assignments with the same distribution as the joint distribution represented by the network

P(C) = 0.5

C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20



S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

Sampling

C	S	R	W
T			

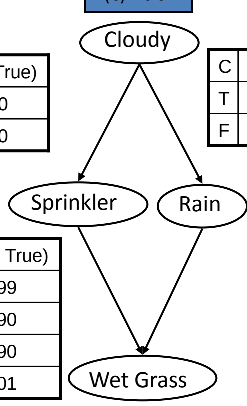
1. Randomly choose C. True with probability 0.5
   → C = True

C =

P(C) = 0.5

C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20



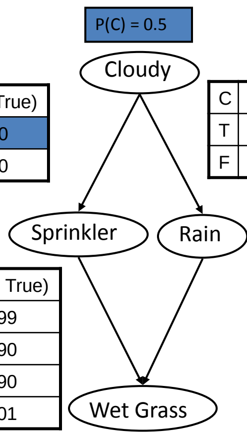
S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

Sampling

C	S	R	W
T	F		

1. Randomly choose C. C =
   True with probability 0.5
   → C = True

2. Randomly choose S. S =
   True with probability 0.10
   → S = False



C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

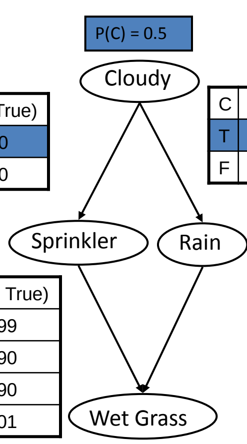
Sampling

C	S	R	W
T	F	T	

1. Randomly choose C. C =
   True with probability 0.5
   → C = True

2. Randomly choose S. S =
   True with probability 0.10
   → S = False

3. Randomly choose R. R =
   True with probability 0.80
   → R = True



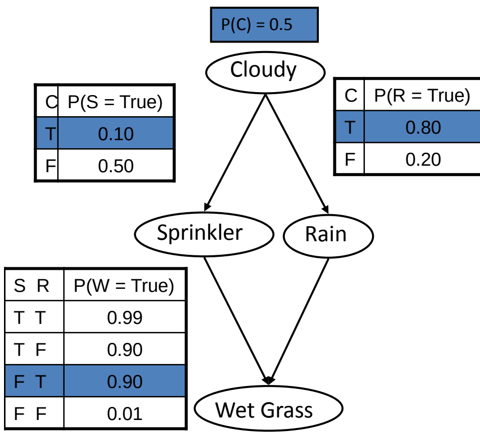
C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

Sampling

C	S	R	W
T	F	T	T



C	P(S = True)
T	0.10
F	0.50

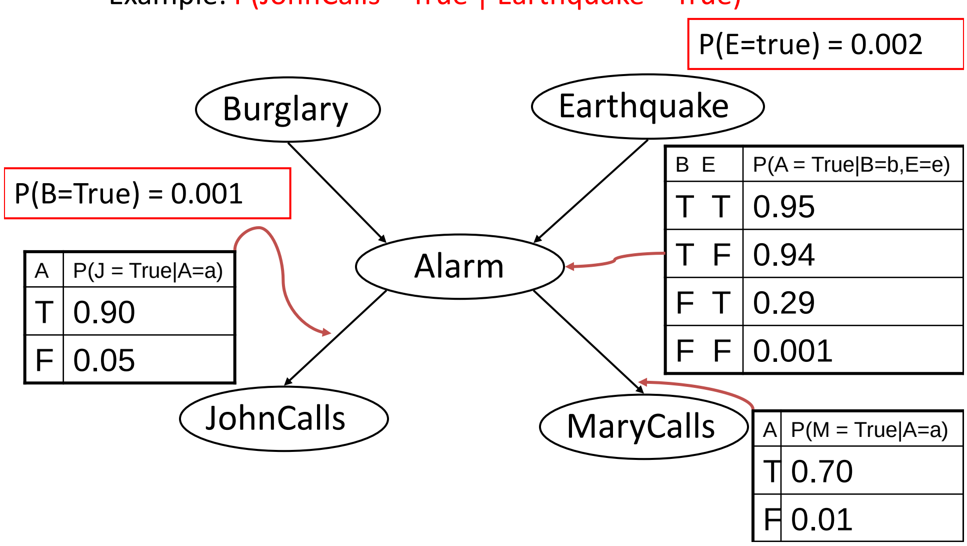
C	P(R = True)
T	0.80
F	0.20

S R	P(W = True)
T T	0.99
T F	0.90
F T	0.90
F F	0.01

1. Randomly choose C. C =
   True with probability 0.5
   → C = True
2. Randomly choose S. S =
   True with probability 0.10
   → S = False
3. Randomly choose R. R =
   True with probability 0.80
   → R = True
4. Randomly choose W. W =
   True with probability 0.90
   → W = True

Problem with Sampling

- Probability is so low for some assignments of variables that that will likely never be seen in the samples (unless a very large number of samples is drawn).
- Example: P(JohnCalls = True | Earthquake = True)



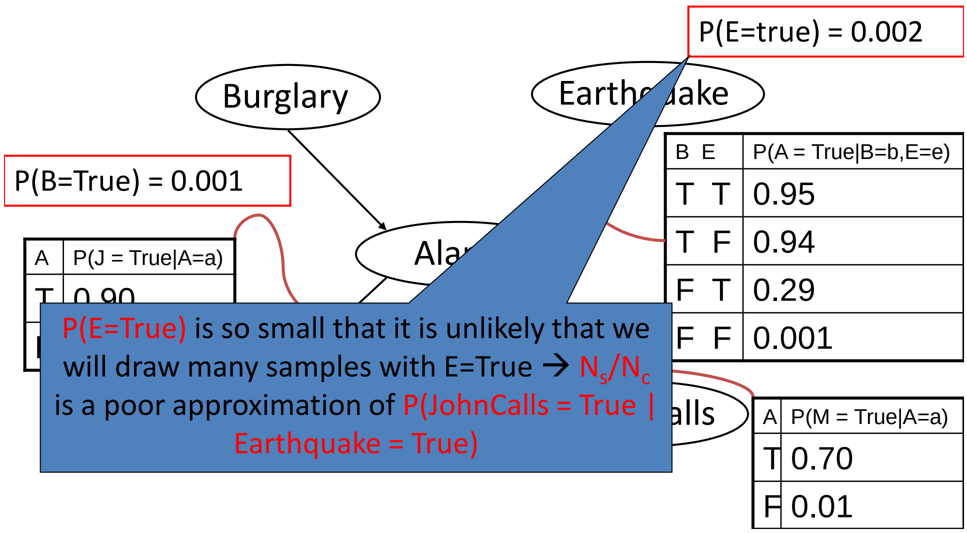
B E	P(A = True|B=b,E=e)
T T	0.95
T F	0.94
F T	0.29
F F	0.001

A	P(J = True|A=a)
T	0.90
F	0.05

A	P(M = True|A=a)
T	0.70
F	0.01

Problem with Sampling

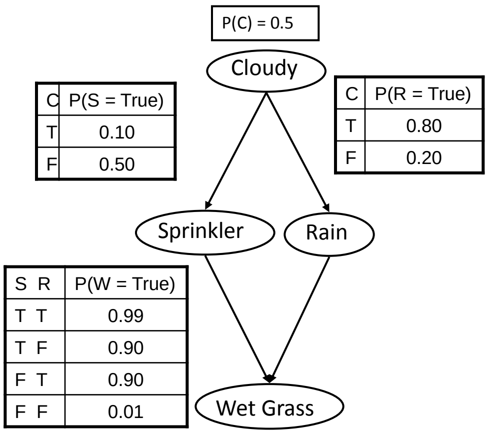
- Probability is so low for some assignments of variables that that they will likely never be seen in the samples (unless a very large number of samples is drawn).
- Example: P(JohnCalls = True | Earthquake = True)



Solution: Likelihood Weighting

- Suppose that E_2 contains a variable assignment of the form $X_i = v$
- Current approach:
  - Generate samples until enough of them contain $X_i = v$
  - Such samples are generated with probability
  - $p = P(X_i = v \mid Parents(X_i))$
- Likelihood Weighting:
  - Generate only samples with $X_i = v$
  - *Reject* samples with $X_i != v$
  - *Weight* each sample by $\omega = p$

Likelihood Weighting



C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

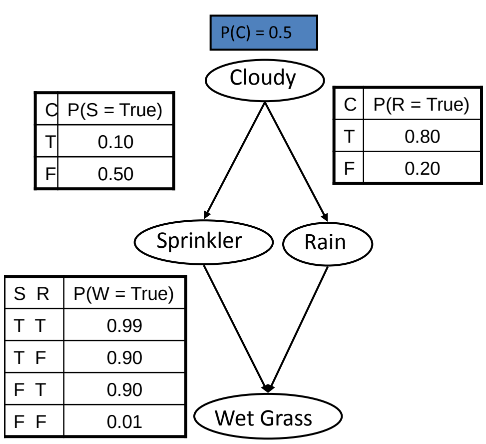
Example: Suppose that we want to compute an inference with

E_2 = (Sprinkler = True , Wet Grass = True)

Likelihood Weighting

ω = 1.0

1. Randomly choose C. C = True with probability 0.5
   → C = True



C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

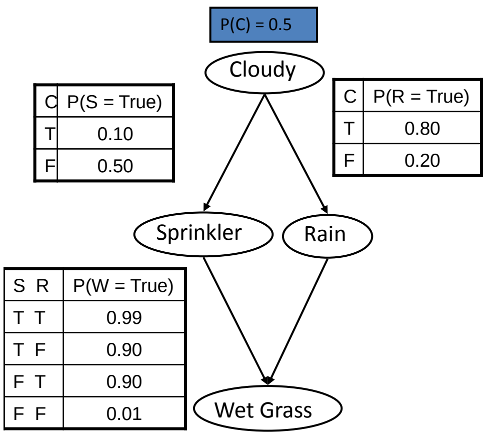
S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

Likelihood Weighting

$\omega = 1.0$

1. Randomly choose C. C = True with probability 0.5 → C = True

C is not one of the evidence variables, so we take a random sample as before



C	P(S = True)
T	0.10
F	0.50

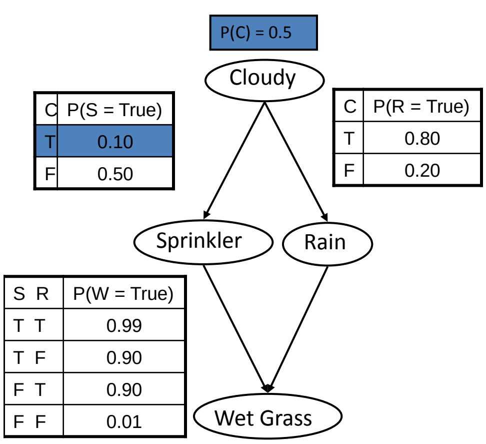
C	P(R = True)
T	0.80
F	0.20

S R	P(W = True)
T T	0.99
T F	0.90
F T	0.90
F F	0.01

Likelihood Weighting

$\omega = 1.0 \times 0.10$

1. Randomly choose C. C = True with probability 0.5 → C = True
2. Set S = True



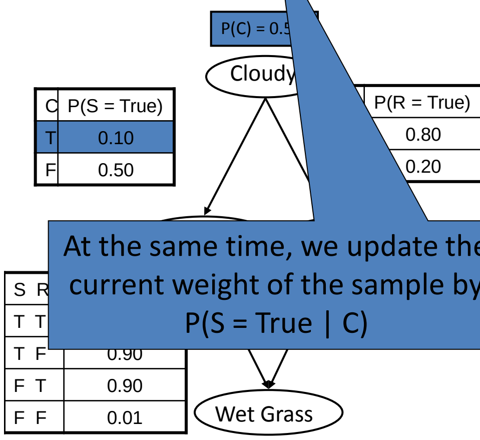
C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

S R	P(W = True)
T T	0.99
T F	0.90
F T	0.90
F F	0.01

Likelihood Weighting

$\omega = 1.0 \times 0.10$



C	P(S = True)
T	0.10
F	0.50

	P(R = True)
	0.80
	0.20

S	R	
T	T	
T	F	0.90
F	T	0.90
F	F	0.01

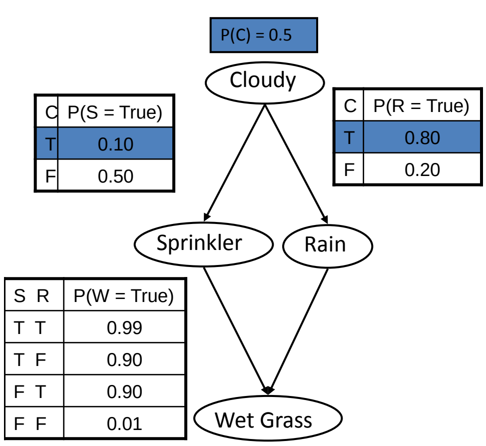
At the same time, we update the current weight of the sample by P(S = True | C)

1. Randomly choose C.
   C = True with probability 0.5 → C = True
2. Set S = True

S is one of the evidence variables, so we fix its value *without* sampling

Likelihood Weighting

$\omega = 1.0 \times 0.10$



C	P(S = True)
T	0.10
F	0.50

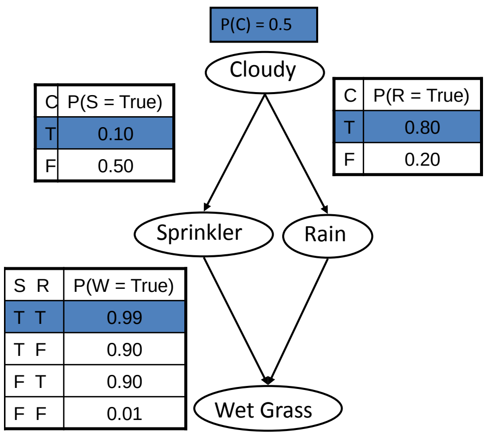
C	P(R = True)
T	0.80
F	0.20

S	R	P(W = True)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

1. Randomly choose C.
   C = True with probability 0.5 → C = True
2. Set S = True
3. Randomly choose R.
   R = True with probability 0.80 → R = True

Likelihood Weighting

ω = 1.0 x 0.10 x 0.99



C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	0.80
F	0.20

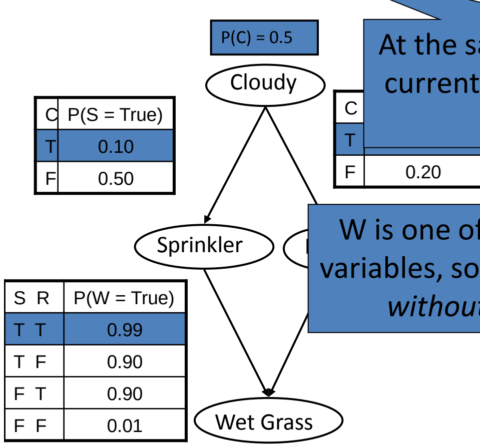
S R	P(W = True)
T T	0.99
T F	0.90
F T	0.90
F F	0.01

1. Randomly choose C. C = True with probability 0.5 → C = True
2. Set S = True
3. Randomly choose R. R = True with probability 0.80 → R = True
4. Set W = True

Likelihood Weighting

ω = 1.0 x 0.10 x 0.99

At the same time, we update the current weight of the sample by P(W = True | S,R)



C	P(S = True)
T	0.10
F	0.50

C	P(R = True)
T	
F	0.20

S R	P(W = True)
T T	0.99
T F	0.90
F T	0.90
F F	0.01

W is one of the evidence variables, so we fix its value *without* sampling

1. Randomly choose C. C = True with probability 0.5 → C =

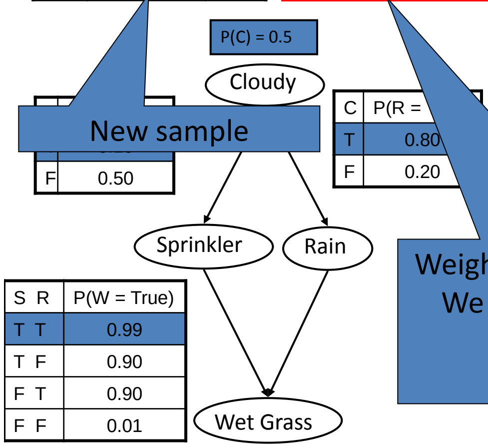
y choose R.
with probability
→
rue

4. Set W = True

Likelihood Weighting

C	S	R	W
T	T	T	T

w = 0.099



1. Randomly choose C.
   C = True with probability 0.5 → C = True
2. Set S = True
3. ...ndomly choose R. ...ty →

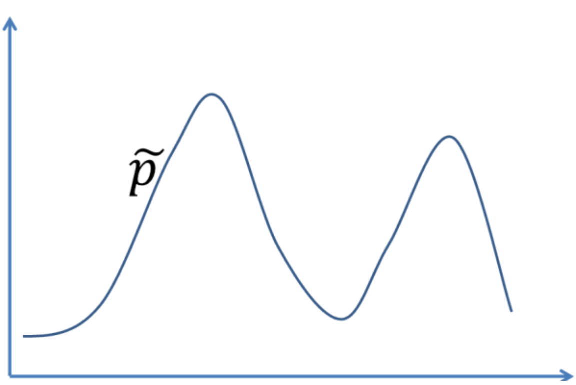
Weight associated with the sample. We increment the "number" of samples by ω:
$N_c \leftarrow N_c + \omega$

4. Set W = True

- Two lessons: Get closer faster to target distribution by
  - Rejecting samples that are not helpful
  - Weighting the samples based on importance
- Assumption:
  - $p(x)$ is impossible to compute but $\tilde{p}(x)$ can be computed:

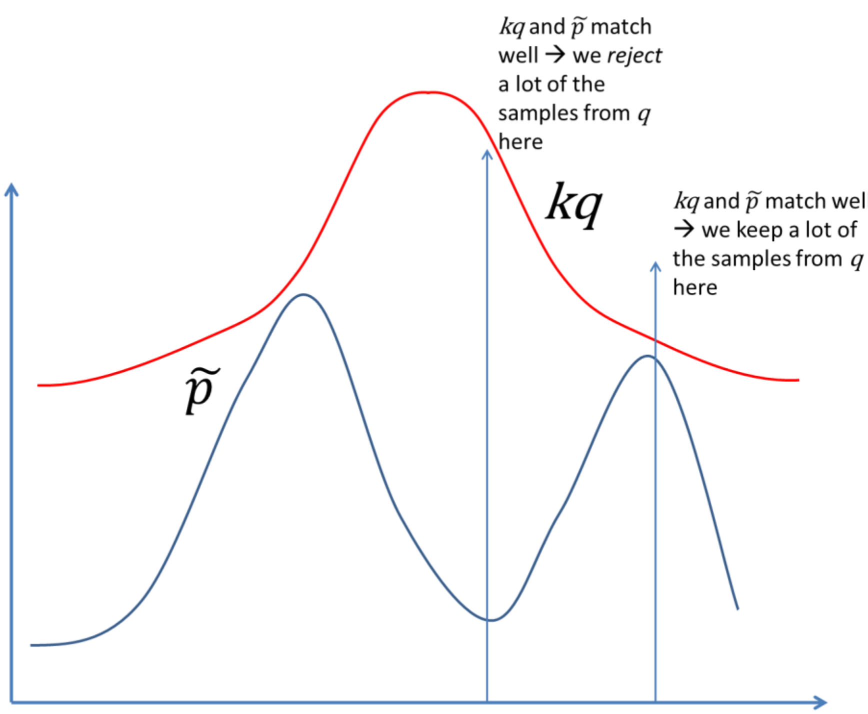
$$p(x) = \frac{1}{Z}\tilde{p}(x)$$

  - Gets around the normalization issue



Rejection

- Proposal distribution (simple): $kq(x) \geq \tilde{p}(x)$
  1. Generate x from $q(.)$
  2. Generate u from $U[0\ \ kq(x)]$
  3. Reject sample if $u > \tilde{p}(x)$
- The closer q is to $\tilde{p}$ the lower the rate of rejection because p(reject) = $1 - \frac{1}{k}\int \tilde{p}$



Importance

- Again "simple" proposal distribution q
- Bad approximation because we can't sample from p directly:
- $E_P(f) = \int f(x)p(x) \approx \frac{1}{N}\sum_i f(x_i)$
- $E_P(f) = \int f(x)\,\frac{p(x)}{q(x)}q(x) \approx \frac{1}{N}\sum_i f(x_i)\,\frac{p(x_i)}{q(x_i)}$

If x_i are sampled from q

"Importance" of x_i

Importance

- p is not normalized so instead:

$$E_P(f) = \frac{1}{Z}\int f(x)\,\frac{\tilde{p}(x)}{q(x)}q(x) \approx \frac{1}{Z}\frac{1}{N}\sum_i f(x_i)\frac{\tilde{p}(x_i)}{q(x_i)}$$

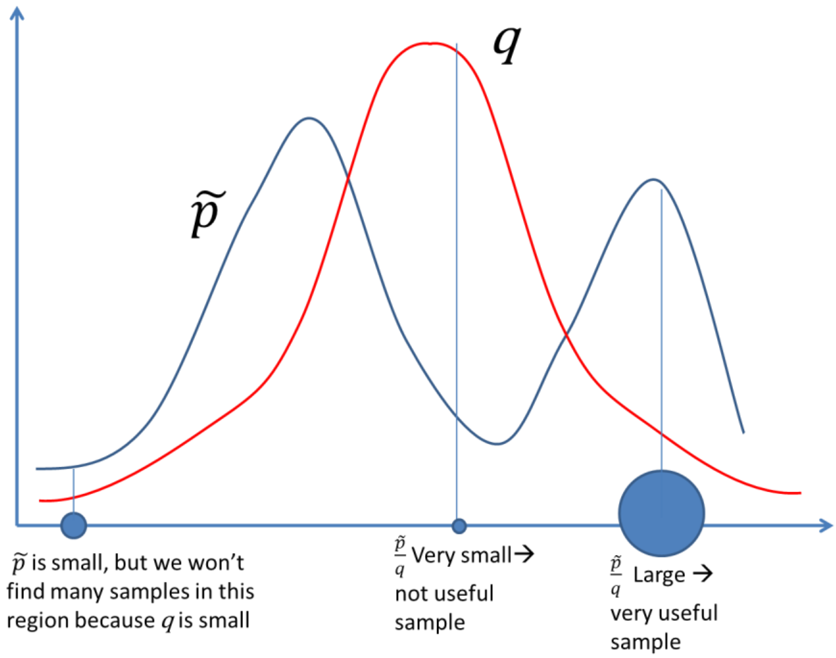
For *f* = 1: $\frac{1}{Z}\frac{1}{N}\sum_i \frac{\tilde{p}(x_i)}{q(x_i)} = 1$

- $E_P(f) \approx$

$$\sum_i f(x_i)w_i \quad w_i = \frac{\tilde{p}(x_i)}{q(x_i)}/\left(\sum_l \tilde{p}(x_l)/q(x_l)\right)$$

If x_i are sampled from q

"Importance" of x_i



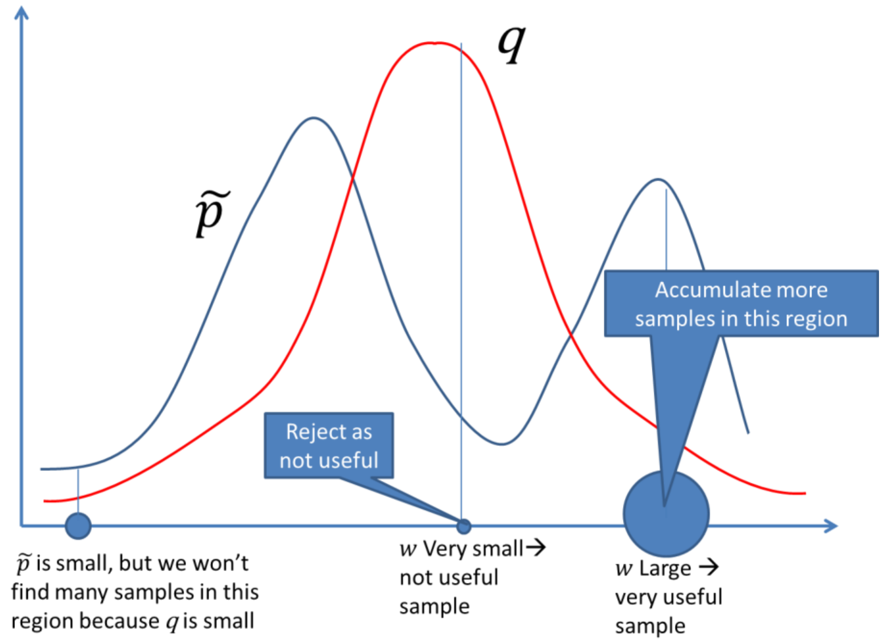
Compromise: SIR

- Fine to evaluate expectation but we may want to draw actual samples

- Draw N samples x_i, w_i (with normalized w_i)
- Draw again N samples from $(x_1, .., x_N)$

using distribution $(w_1, .., w_N)$

- Basically: Smart way of reject samples with low weight
- Guaranteed to converge to p when $N \rightarrow \infty$



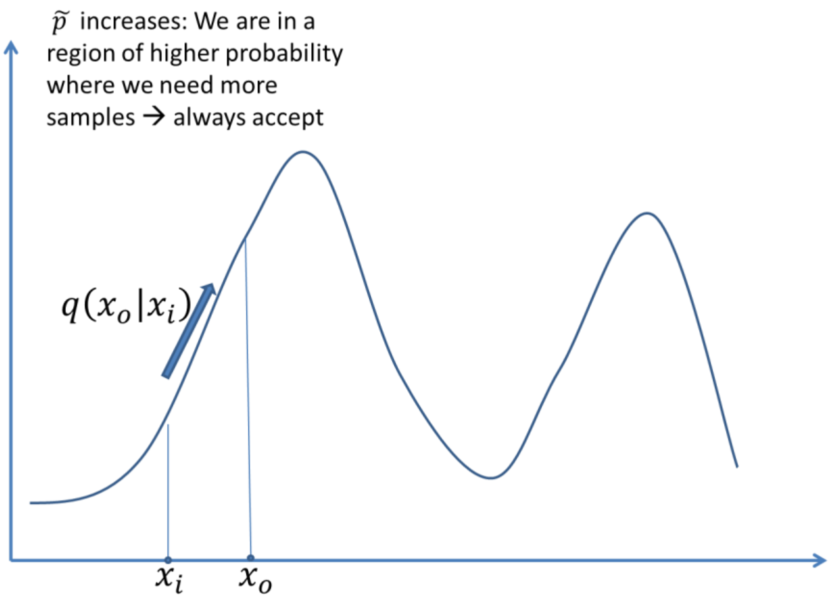
Adapting to the sampled distribution

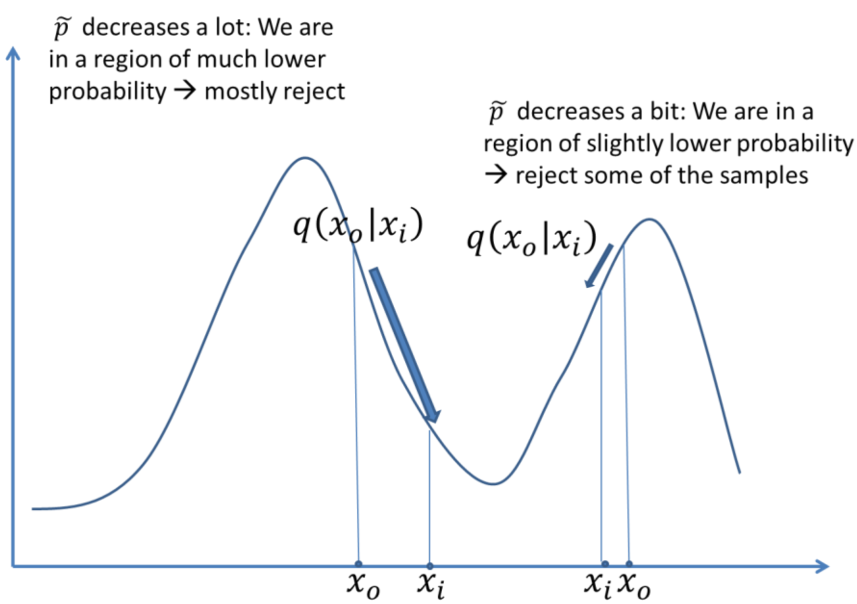
- Problem: The proposal distribution q might be arbitrarily bad relative to p
- Idea (Metropolis): Adapt to the local shape of p

1. Condition the choice of a sample on the previous sample $q(x_o|x_i)$ $(q(a,b) = q(b,a) > 0)$
2. Accept if $\tilde{p}(x_o) > \tilde{p}(x_i)$
3. With probability $\frac{\tilde{p}(x_o)}{\tilde{p}(x_i)}$ otherwise

Still guaranteed to converge, but samples are not independent

Very inefficient because q may not be "adapted" to p


$\tilde{p}$ increases: We are in a region of higher probability where we need more samples → always accept
$q(x_o|x_i)$
x_i
x_o


$\tilde{p}$ decreases a lot: We are in a region of much lower probability → mostly reject
$\tilde{p}$ decreases a bit: We are in a region of slightly lower probability → reject some of the samples
$q(x_o|x_i)$
$q(x_o|x_i)$
x_o
x_i
x_i
x_o

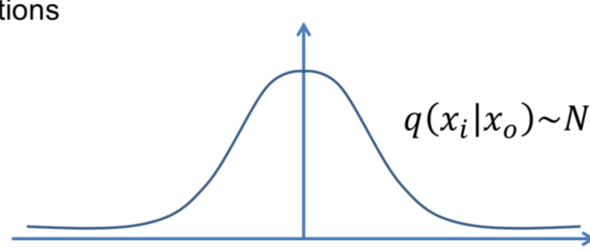
Better adaptation

- Same idea but with not requiring symmetric q

1. Accept if $\tilde{p}(x_o) > \tilde{p}(x_i)$
2. With probability $\frac{\tilde{p}(x_o)}{\tilde{p}(x_i)} \frac{q(x_i|x_o)}{q(x_o|x_i)}$ otherwise

Multiple q_k can be used

Example q:
Small σ = Takes small steps (random walk)
Large σ = Faster exploration of the space but lots more rejections



$q(x_i|x_o) \sim N(x_o, \sigma)$

MH = Metropolis Hastings

Why does it work?

- The samples $x_1, .., x_t$ are such that the probability of choosing a sample at time t+1 depends only on the previous sample:

$$\bar{p}(x_{t+1}) = \sum_{x_t} p(x_{t+1}|x_t)\bar{p}(x_t)$$

- $\bar{p}$ converges to a distribution p if:

$$p(x)T(x, x') = p(x')T(x', x)$$

- Sufficient condition (reversibility):
  - It turns out that $\min(1, \frac{\tilde{p}(x_o)}{\tilde{p}(x_i)} \frac{q(x_i|x_o)}{q(x_o|x_i)})$ satisfies this condition
  - Distribution of x_t converges to p

Caveats

- Burn-in: Takes a (unknown, possible long) amount of time to converge to p
- Selection of q: Compromise between moving fast through space and not rejecting too many samples

Back to sampling from joint distribution

- We want to sample from $p(X) = p(x_1, .., x_n)$
- Assume that it's easy to sample from: $q_k(x) = p(x|X_{\backslash \mathrm{k}})$
- Use q_k as n proposals used in turn
- Turns out that these proposals are *always* accepted

- Gibbs sampling (step i):

Sample x_{1i+1} from $p(x|x_{2i}, .., x_{ni})$

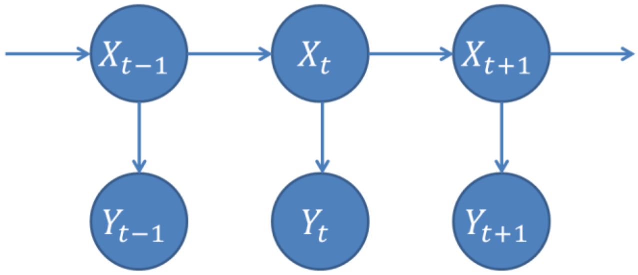
……….

Sample x_{ki+1} from $p(x|x_{1i+1}, .., x_{k-1i+1}, x_{k+1i}, .., x_{ni})$

……..

Sample x_{ni+1} from $p(x|x_{1i+1}, .., x_{n-1i+1})$

Sequential models



- Interesting cases:
- $P(x_{t+1}|x_t)$ hard → Need to sample
- $P(y_t|x_t)$ "easy" to evaluate for a given x_t
- Localization:
  $x = [u\ v\ \theta]$ complex banana transition distribution
  Given position/orientation: Can compute measurements
- Tracking:
  x = positions and orientations of many joints → Very non-linear; hard to manipulate but can be sampled

Sequential models

$$p(x_{t+1}|y_{1:t+1}) \alpha p(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{1:t})$$

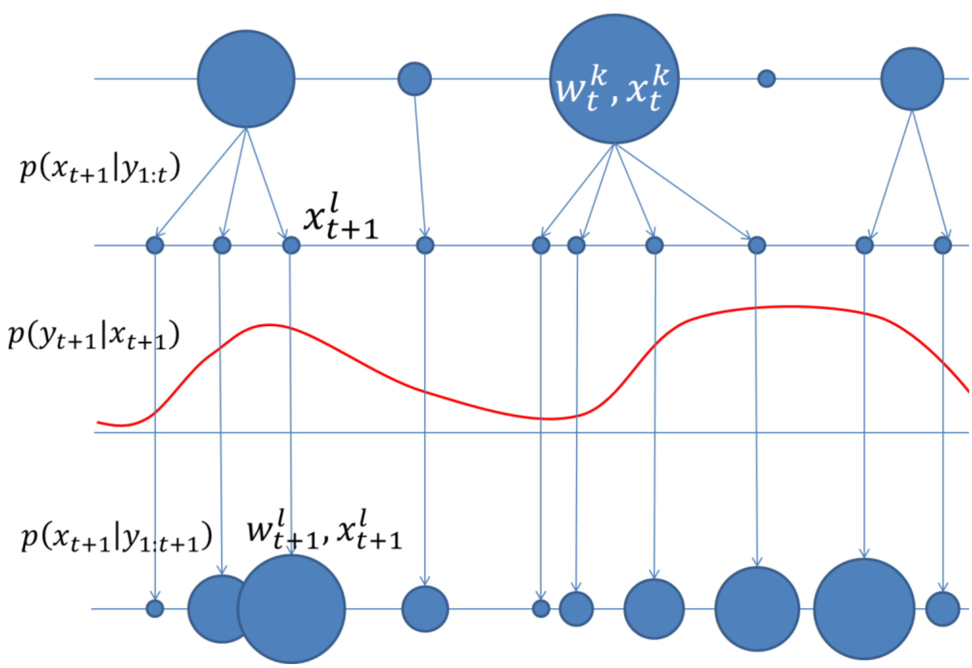
$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t) p(x_t|y_{1:t}) dx_t$$

Suppose that we have K samples w_t^k, x_t^k describing the distribution at the previous time step:

$$p(x_{t+1}|y_{1:t}) \approx \sum_k w_t^k p(x_{t+1}|x_t^k)$$

Example particle filter

- $L = 1,..,K$:
  1. Sample x_{t+1}^{l} from $p(x_{t+1}|y_{1:t}) \approx \sum_k w_t^k p(x_{t+1}|x_t^k)$:
     a. Pick a sample x_t^l using the distribution $(w_t^1, \ldots, w_t^K)$
     b. Sample x_{t+1}^l from $p(x_{t+1}|x_t^l)$
  2. Assign weight $w_{t+1}^l = p(y_{t+1}|x_{t+1}^l)$
  3. Normalize w_{t+1}^l



Sampling

- First (simple and silly) example on a couple of Bayes nets
  - Ancestral and likelihood sampling
- General techniques
  - Rejection
  - Importance
  - MCMC
  - Gibbs
  - Sequential (particles)