15-780: Graduate AI
Lecture 15. Uncertainty, Inference

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Sam Ganzfried, Byron Boots

Review

Spatial planning

- *C-space*
- *Ways of splitting up C-space*
  - *Visibility graph*
  - *Voronoi*
  - *Cell decomposition*
  - *Variable resolution or adaptive cells (quadtree, parti-game)*

RRTs

- *Build tree by randomly picking a point, extending tree toward it*
- *Optionally:*
    - *build forward and backward at once*
    - *cross-link within tree*
- *Plan within tree to get close enough to goal*

RRTs

- *Tend to break up large Voronoi regions*
- *So, RRT search is coarse to fine*
- *First path found usually suboptimal; if we continue to grow tree and cross-link, get optimality in limit*

Probability

- *Random variables*
- *Events (atomic, composite, AND/OR/NOT)*
- *Distributions: joint, marginal*
- *Law of total probability*
  - $P(X) = P(X, Y=y_1) + P(X, Y=y_2) + \ldots$

Probability

Conditional: incorporating observations

		15-780		
		A	*A-*	*B+*
HW4	*A*	*.21*	*.17*	*.07*
	A-	*.17*	*.15*	*.06*
	B+	*.07*	*.06*	*.04*

15-780	*A*	*.41*
	A-	*.35*
	B+	*.24*

$P(15\text{-}780=A \mid HW4=B+) = .04 / (.07+.06+.04)$

Conditioning

- *In general, divide a row or column by sum*
  - $P(X \mid Y=y) = P(X, Y=y) / P(Y=y)$
- $P(Y=y)$ *is a marginal probability*
- $P(X \mid Y=y)$ *is a row or column of table*
- *Thought experiment: what happens if we condition on an event of zero probability?*

Notation

- $P(X \mid Y)$ is a function: $x, y \rightarrow P(X=x \mid Y=y)$
- As is standard, expressions are evaluated separately for each realization:
  - $P(X \mid Y)\,P(Y)$ means the function $x, y \rightarrow P(X=x \mid Y=y)\,P(Y=y)$

Independence

- *X and Y are* ***independent*** *if, for all possible values of y,* $P(X) = P(X \mid Y{=}y)$
  - *equivalently, for all possible values of x,* $P(Y) = P(Y \mid X{=}x)$
  - *equivalently,* $P(X, Y) = P(X)\,P(Y)$
- *Knowing X or Y gives us no information about the other*

Independence: probability = product of marginals

AAPL price

Weather

	up	*same*	*down*	
sun	*.09*	*.15*	*.06*	*0.3*
rain	*.21*	*.35*	*.14*	*0.7*
	0.3	*0.5*	*0.4*	

Bayes Rule

- *For any X, Y, C*
  - $P(X \mid Y, C)\, P(Y \mid C) = P(Y \mid X, C)\, P(X \mid C)$
- *Simple version (without context)*
  - $P(X \mid Y)\, P(Y) = P(Y \mid X)\, P(X)$
- *Proof: both sides are just* $P(X, Y)$
  - $P(X \mid Y) = P(X, Y) / P(Y)$ *(by def'n of conditioning)*

Continuous distributions

- *What if X is real-valued?*
  - *Atomic event:* $(X \leq x)$, $(X \geq x)$
    - *any* other subset via AND, OR, NOT*
  - $X=x$ *has zero* probability*
- $P(X=x, Y=y)$ *means*
  - $\lim P(x \leq X \leq x+\varepsilon, Y = y) / \varepsilon \qquad \varepsilon \to 0+$

Factor Graphs

Factor graphs

- *Strict generalization of SAT to include uncertainty*
- *Factor graph = (variables, constraints)*
  - *variables: set of discrete r.v.s*
  - *constraints: set of (possibly)* ***soft*** *or* ***probabilistic*** *constraints called* ***factors*** *or* ***potentials***

Factors

- *Hard constraint:* $X + Y \geq 3$
- *Soft constraint:* $X + Y \geq 3$ *is more probable than* $X + Y < 3$*, all else equal*
- *Domain of factor: set of relevant variables*
    - $\{X, Y\}$ *in this case*

Factors

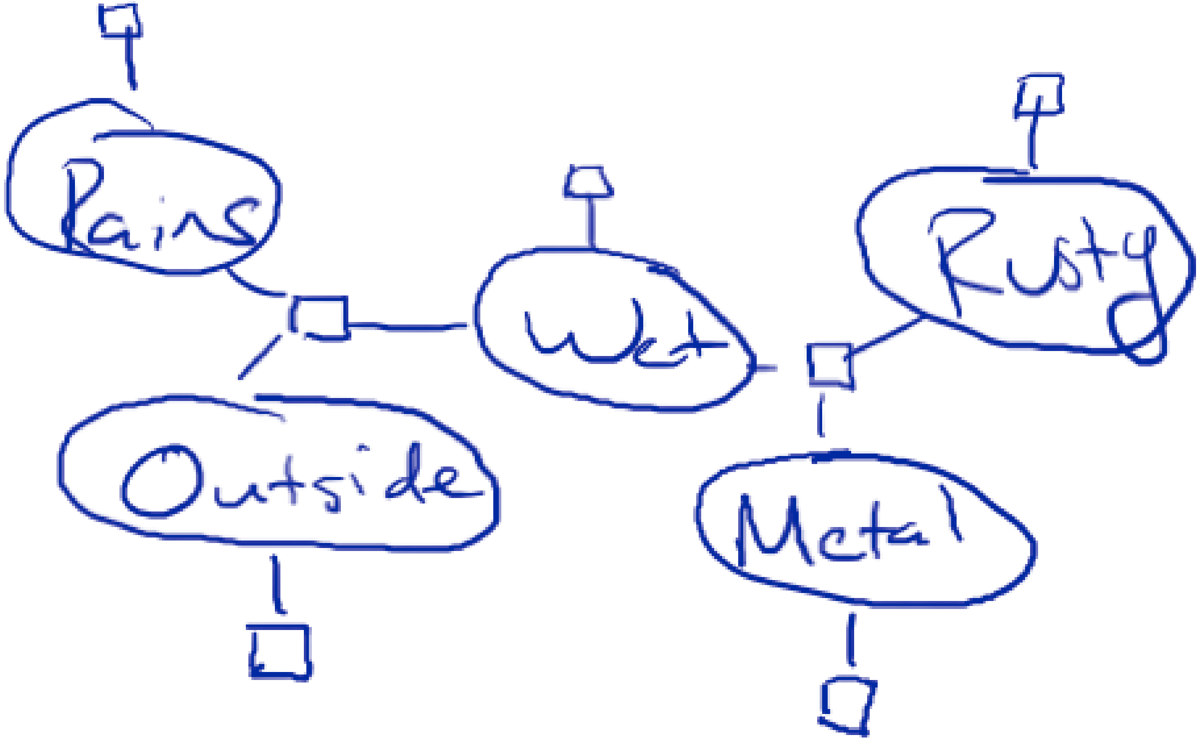
Hard

		X		
		0	1	2
	0	0	0	0
Y	1	0	0	1
	2	0	1	1

Soft

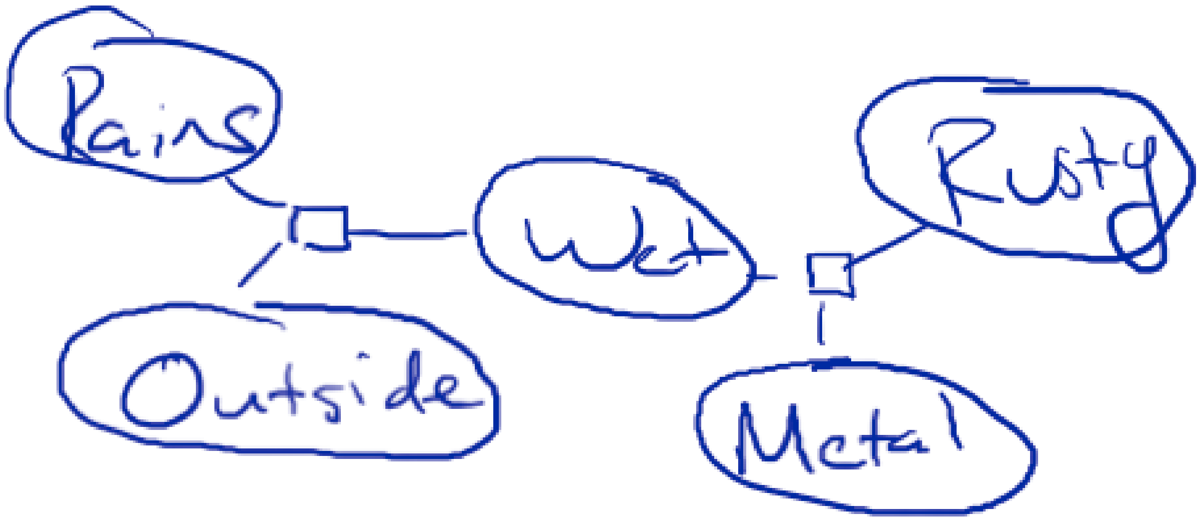
		X		
		0	1	2
	0	1	1	1
Y	1	1	1	3
	2	1	3	3

Factor graphs



- *Variables and factors connected according to domains*

Factor graphs



- *Omit single-argument factors from graph to reduce clutter*

Factor graphs: probability model

$$P(Ra, W, O, M, Ru) = \phi(Ra, W, O, M, Ru)/Z$$

$$\phi(Ra, W, O, M, Ru) = \phi_1(Ra, W, O)\phi_2(O, M, Ru) \phi_3(Ra)\phi_4(W)\phi_5(O) \phi_6(M)\phi_7(Ru)$$

Normalizing constant

$$Z = \sum_{Ra} \sum_{W} \sum_{O} \sum_{M} \sum_{Ru} \phi(Ra, W, O, M, Ru)$$

- *Also called* ***partition function***

Factors

Ra	O	W	Φ_1
T	T	T	3
T	T	F	1
T	F	T	3
T	F	F	3
F	T	T	3
F	T	F	3
F	F	T	3
F	F	F	3

W	M	Ru	Φ_2
T	T	T	3
T	T	F	1
T	F	T	3
T	F	F	3
F	T	T	3
F	T	F	3
F	F	T	3
F	F	F	3

Unary factors

Ra	Φ_3
T	1
F	2

W	Φ_4
T	1
F	1

O	Φ_5
T	5
F	2

M	Φ_6
T	10
F	1

Ru	Φ_7
T	1
F	3

Inference Qs

- *Is* $Z > 0$*?*
- *What is* $P(E)$*?*
- *What is* $P(E_1 \mid E_2)$*?*
- *Sample a random configuration according to* $P(.)$ *or* $P(. \mid E)$
- *Hard part: taking sums of* Φ *(such as* Z*)*

Example

- *What is* $P(T, T, T, T, T)$*?*

Example

- *What is P(Rusty=T | Rains=T)?*
- *This is P(Rusty=T, Rains=T) / P(Rains=T)*
- *P(Rains) is a marginal of P(…)*
- *So is P(Rusty, Rains)*
- *Note: Z cancels, but still have to sum lots of entries to get each marginal*

Relationship to SAT, ILP

- *Easy to write a clause or a linear constraint as a factor: 1 if satisfied, 0 o/w*
- *Feasibility problem: is Z > 0?*
  - *more generally, count satisfying assignments (determine Z)*
  - *NP or #P complete (respectively)*
- *Sampling problem: return a satisfying assignment uniformly at random*

Inference

Notation

- *Boldface = vector of r.v.s or values*
  - $\boldsymbol{X} = (X_1, X_2, X_3, X_4, X_5)$
  - $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$
- *Set index = subvector*
  - *If* $D = \{1, 3, 4\}$ *then* $\mathbf{X}_D = (X_1, X_3, X_4)$
- *In particular,* $\mathbf{X}_{D(\Phi)}$ *= input to factor* $\boldsymbol{\Phi}$

Finding **x** w/ P(**X**=**x**)>0

- $\{\mathbf{x} \mid P(\mathbf{X}=\mathbf{x})>0\}$ = ***support*** *of* $P(\mathbf{X})$

- *Replace each factor* Φ *with* $\Phi' = I(\Phi>0)$
- *Now we have a CSP (or SAT); do DPLL*

Counting support

- *DPLL also works to count* $\{\mathbf{x} \mid P(\mathbf{X}=\mathbf{x})>0\}$
- *Or, we can get smarter: dynamic programming*
- *Example:* $(A \vee B) \wedge (B \vee C) \wedge (C \vee D)$

DP: $(A \vee B) \wedge (B \vee C) \wedge (C \vee D)$

A	B	Φ_1
T	T	1
T	F	1
F	T	1
F	F	0

B	C	Φ_2
T	T	1
T	F	1
F	T	1
F	F	0

C	D	Φ_3
T	T	1
T	F	1
F	T	1
F	F	0

- *Consider B=T and B=F separately*
- *B=F: 1;   B=T: 2*2 = 4*
- *Subtotal: |supp(ABC)| = 5; note C=F in 2*

DP: (A ∨ B) ∧ (B ∨ C) ∧ (C ∨ D)

A	B	Φ_1
T	T	1
T	F	1
F	T	1
F	F	0

B	C	Φ_2
T	T	1
T	F	1
F	T	1
F	F	0

C	D	Φ_3
T	T	1
T	F	1
F	T	1
F	F	0

- *Consider C=T and C=F separately*
- *C=F: 2*1=2;   C=T: 3*2 = 6*
- *Total: |support| = 8*

Quiz

- *What is size of support for*

  $(A \vee B) \wedge (B \vee C) \wedge (C \vee D) \wedge (D \vee E)$

DP for finding Z

- *Same trick works for finding Z*
  - *not surprising: Z is a sum over support*

Factors

Ra	*O*	*W*	Φ_1
T	*T*	*T*	*3*
T	*T*	*F*	*1*
T	*F*	*T*	*3*
T	*F*	*F*	*3*
F	*T*	*T*	*3*
F	*T*	*F*	*3*
F	*F*	*T*	*3*
F	*F*	*F*	*3*

Ra	Φ_3
T	*1*
F	*2*

O	Φ_5
T	*5*
F	*2*

W	Φ_4
T	*1*
F	*1*

Factors

Ra	O	W	Φ_1	Φ_3	Φ_4	Φ_5	*
T	T	T	3	1	1	5	15
T	T	F	1	1	1	5	5
T	F	T	3	1	1	2	6
T	F	F	3	1	1	2	6
F	T	T	3	2	1	5	30
F	T	F	3	2	1	5	30
F	F	T	3	2	1	2	12
F	F	F	3	2	1	2	12

Ra	Φ_3
T	1
F	2

O	Φ_5
T	5
F	2

W	Φ_4
T	1
F	1

Factors

Ra	*O*	*W*	*
T	*T*	*T*	*15*
T	*T*	*F*	*5*
T	*F*	*T*	*6*
T	*F*	*F*	*6*
F	*T*	*T*	*30*
F	*T*	*F*	*30*
F	*F*	*T*	*12*
F	*F*	*F*	*12*

W	λ
T	*63*
F	*53*

Factors

W	M	Ru	Φ_2	Φ_6	Φ_7	λ	*
T	T	T	3	10	1	63	1890
T	T	F	1	10	3	63	1890
T	F	T	3	1	1	63	189
T	F	F	3	1	3	63	567
F	T	T	3	10	1	53	1590
F	T	F	3	10	3	53	4770
F	F	T	3	1	1	53	159
F	F	F	3	1	3	53	477

M	Φ_6
T	10
F	1

Ru	Φ_7
T	1
F	3

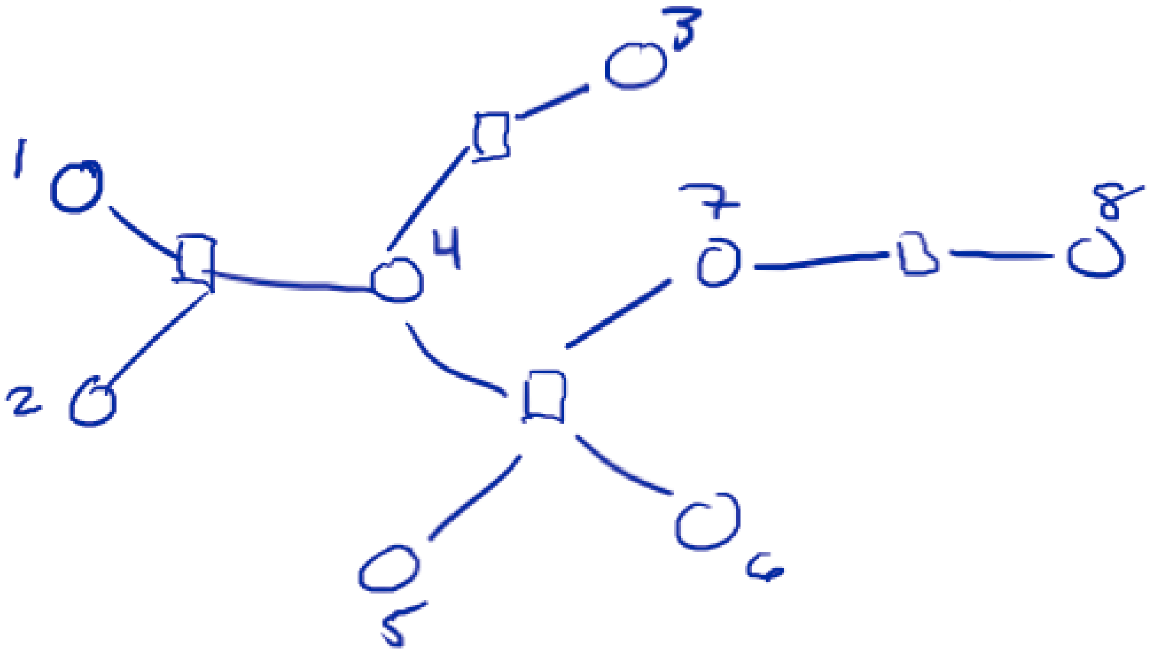
W	λ
T	63
F	53

Factors

W	M	Ru	*
T	T	T	1890
T	T	F	1890
T	F	T	189
T	F	F	567
F	T	T	1590
F	T	F	4770
F	F	T	159
F	F	F	477

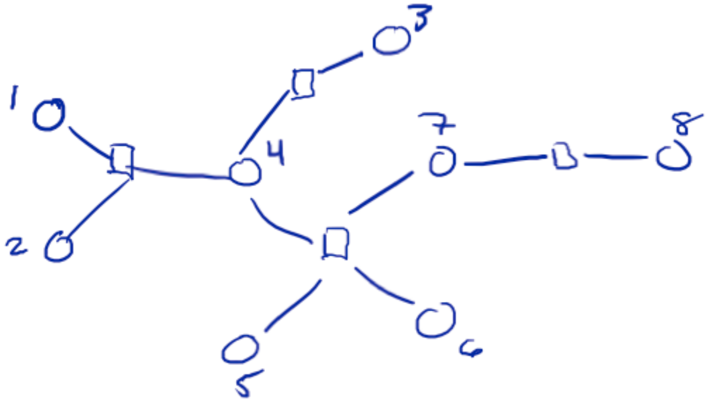
$Z = 11532$

Variable elimination



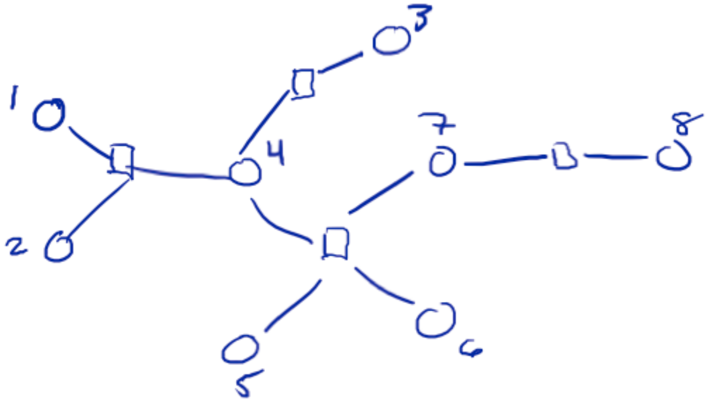
- *Basic step: move a sum inward as far as possible, then do sum for each possible way of setting neighbors*

Eliminating partway



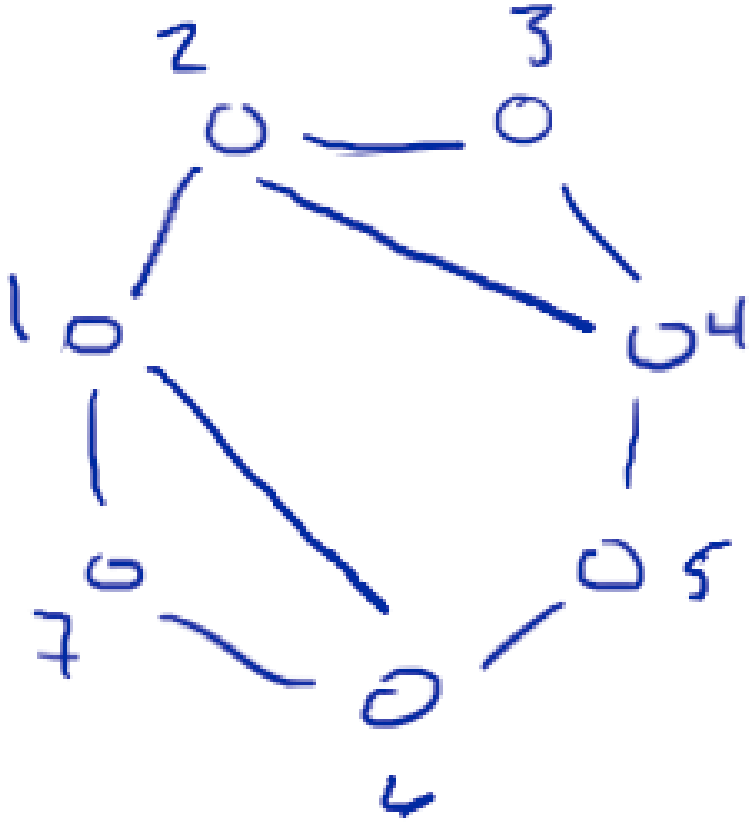
- *For Z, want to sum over all* $\boldsymbol{X}$
- *For marginal* $P(\boldsymbol{X}_{45})$, *eliminate* $\boldsymbol{X}_{123678}$
- *Sum out 123, then 876, to get Z* $P(\boldsymbol{X}_{45})$
- *Sum out 45 to get Z*

Eliminating partway



- Conditional $P(\mathbf{X}_{45} \mid \mathbf{X}_6) = P(\mathbf{X}_{456}) / P(\mathbf{X}_6)$
- Sum out 123, then 87, to get $Z\,P(\mathbf{X}_{456})$
- Sum out 45 to get $Z\,P(\mathbf{X}_6)$
- Divide

A more difficult example



Treewidth

- *Elimination order E: sum x_1, then x_5, …*
- *treewidth(E) =*

  (size of largest factor formed) – 1
- *treewidth = $\min_E$ treewidth(E)*
- *Variable elimination uses space, time* ***exponential*** *in treewidth*
- *Worse: even computing treewidth is NP-complete*

Belief propagation

- *Suppose we want all 1-variable marginals*
- *Could do N runs of variable elimination*
- *Or: the BP algorithm simulates N runs for the price of 2*
- *For details: Kschischang et al. reading*

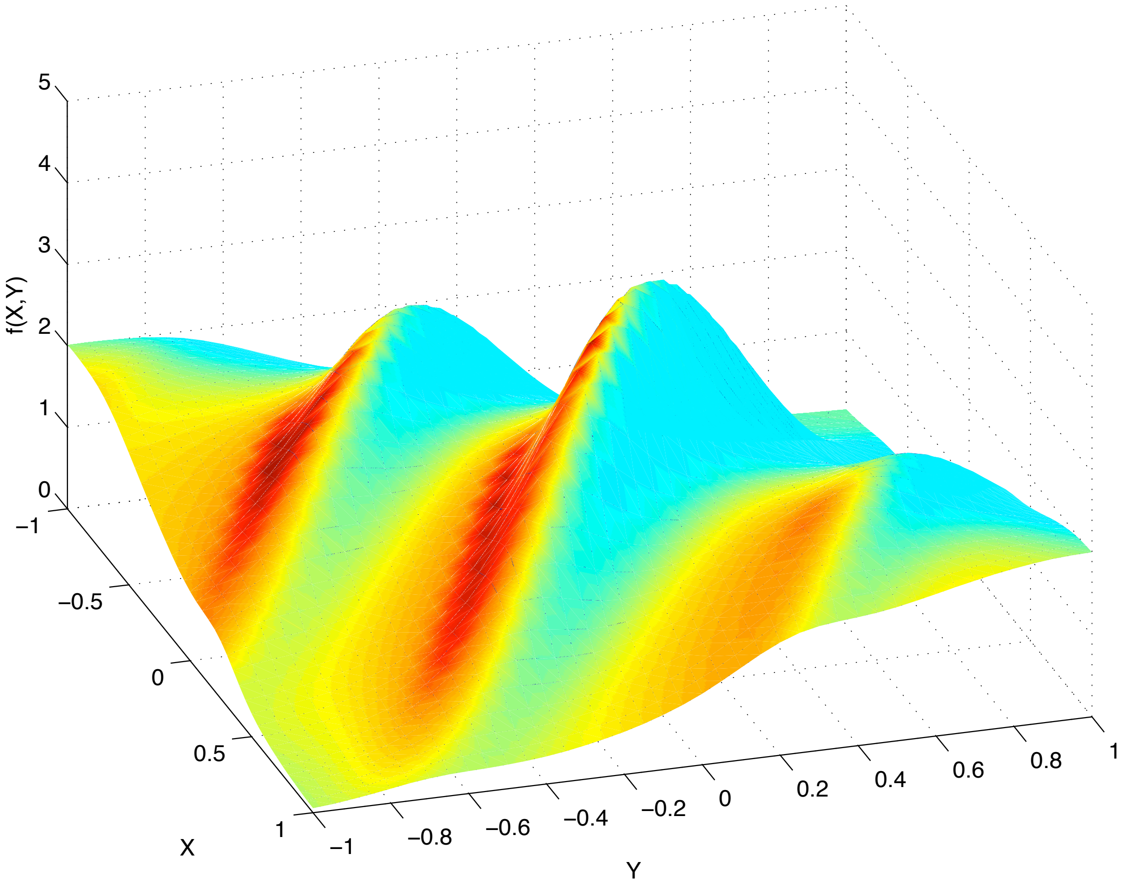
Approximate Inference

Most of the time…

- *Treewidth is big*
- *Variables are high-arity or continuous*
- *Can't afford exact inference*

- *Partition function = numerical integration (and/or summation)*
- *We'll look at randomized algorithms*

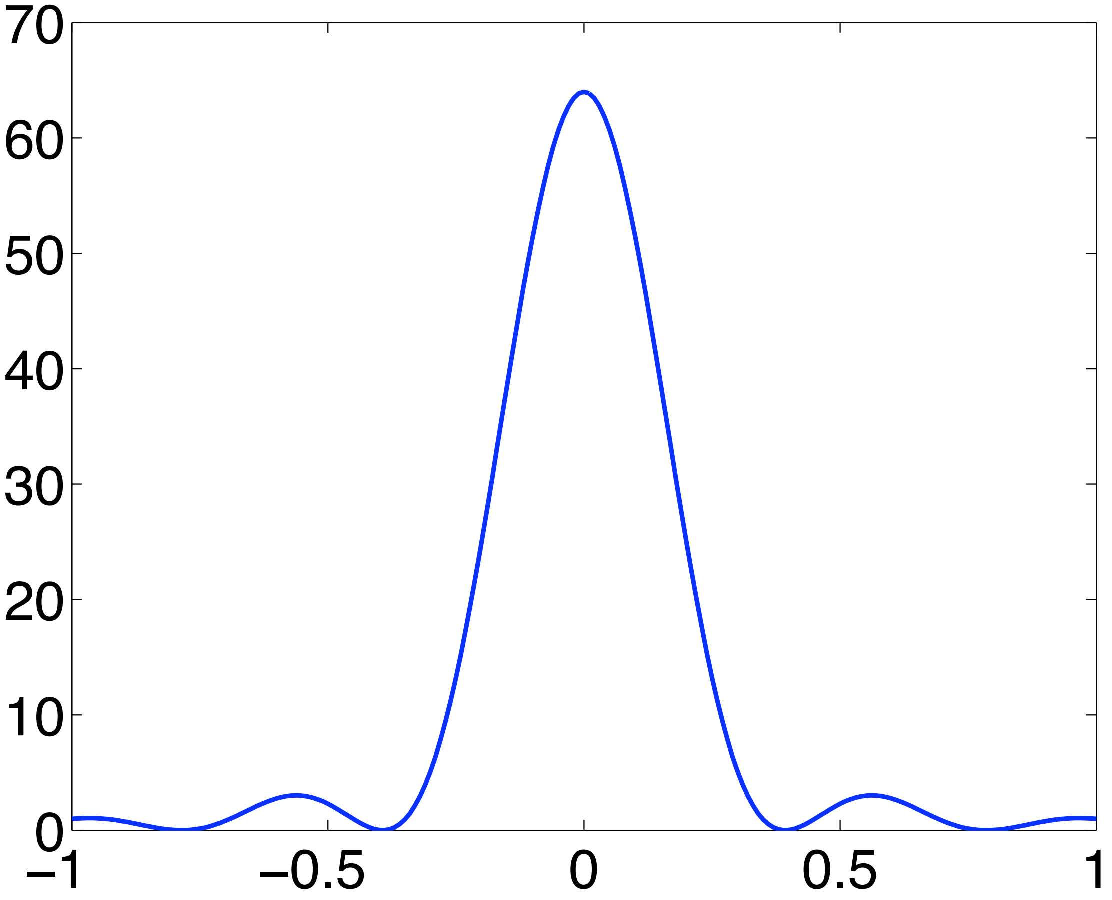
Numerical integration



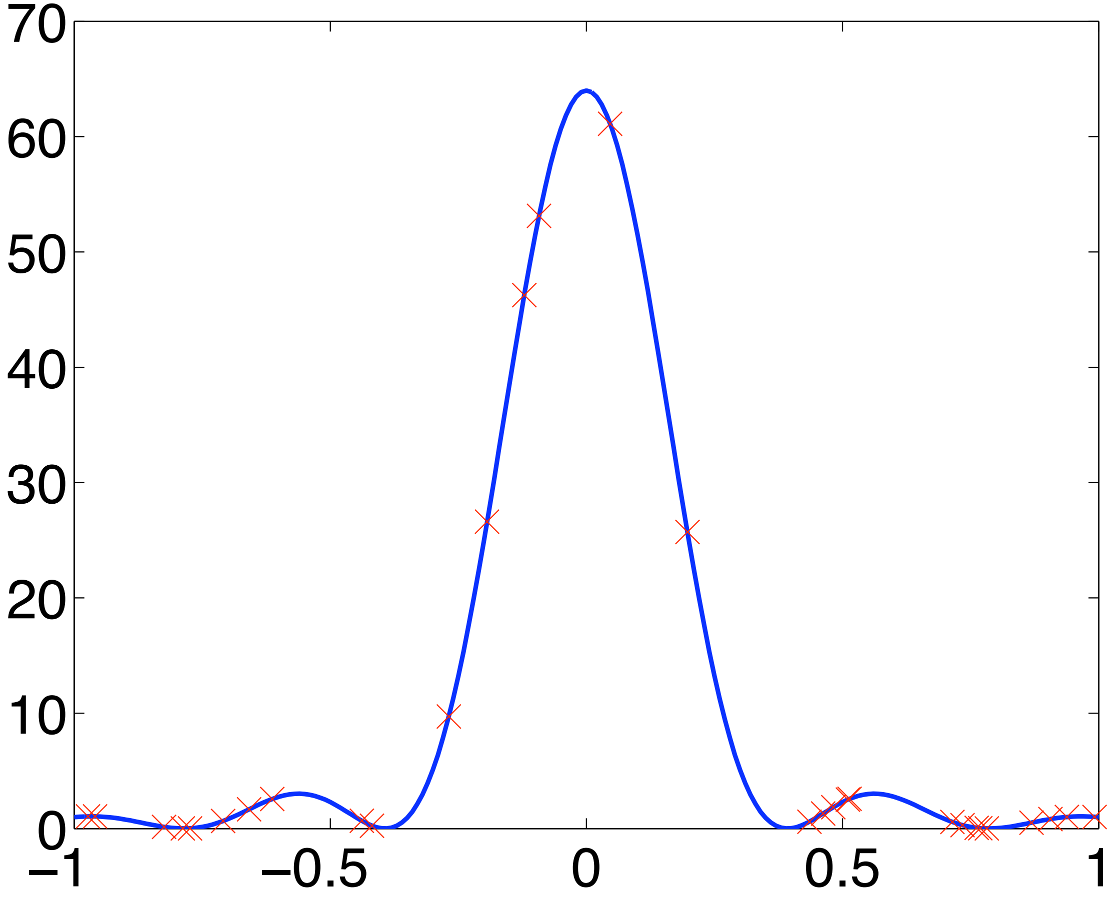
Integration in 1000s of dims

Eliazar and Parr, IJCAI-03

Simple 1D problem



Simplest randomized algorithm



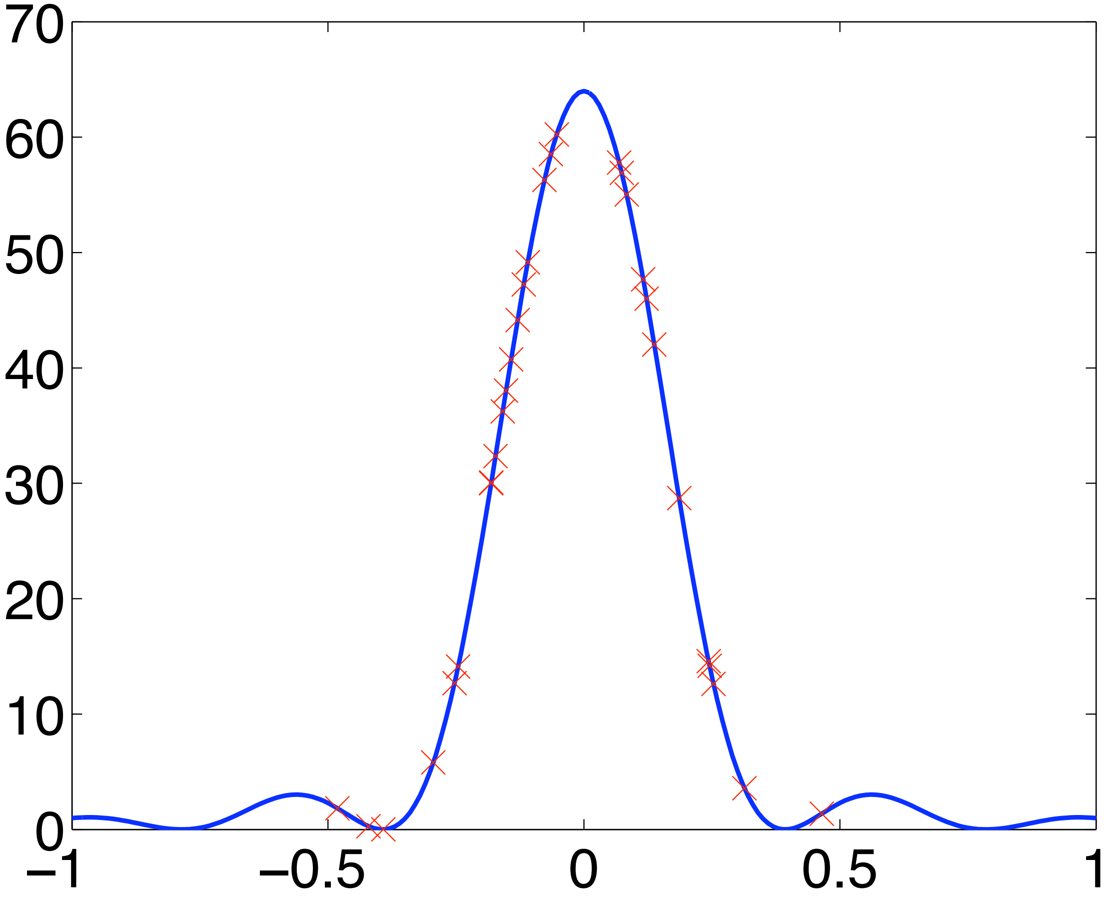
- *Uniform sampling: sum(f(x_i))/N*

Uniform sampling

$$
\begin{aligned}
E(f(X)) &= \int P(x) f(x) dx \\
&= \frac{1}{V} \int f(x) dx
\end{aligned}
$$

- *So, V E(f(X)) is desired integral*
- *But standard deviation can be big*
- *Can reduce it by averaging many samples*
- *But only at rate 1/sqrt(N)*

Nonuniform (importance) sampling



Importance sampling

- *Instead of x ~ uniform, use x ~ Q(x)*
- *Q = importance distribution*
- *Should have Q(x) large where f(x) is large*
- *Problem:*

$$E_Q(f(X)) = \int Q(x)f(x)dx$$

Importance sampling

$$h(x) \equiv f(x)/Q(x)$$

$$\begin{aligned} E_Q(h(X)) &= \int Q(x)h(x)dx \\ &= \int Q(x)f(x)/Q(x)dx \\ &= \int f(x)dx \end{aligned}$$

Importance sampling

- *So, take samples of h(X) instead of f(X)*
- $w_i = 1/Q(x_i)$ *is* ***importance weight***
- Q *= uniform yields uniform sampling*

Variance

- *How does this help us control variance?*
- *Suppose f big ==> Q big*
- *And Q small ==> f small*
- *Then h = f/Q never gets too big*
- *Variance of each sample is lower ==> need fewer samples*
- *A good Q makes a good IS*

Importance sampling, part II

- *Suppose we want*

$$\int f(x)dx = \int P(x)g(x)dx = E_P(g(X))$$

- *Pick N samples x_i from proposal Q(X)*
- *Average w_i g(x_i), where w_i = P(x_i)/Q(x_i) is importance weight*

$$E_Q(Wg(X)) = \int Q(x)[P(x)/Q(x)]g(x)dx = \int P(x)g(x)dx$$

Parallel importance sampling

- *Suppose we want*

$$\int f(x)dx = \int P(x)g(x)dx = E_P(g(X))$$

- *But P(x) is unnormalized (e.g., represented by a factor graph)—know only Z P(x)*

Parallel IS

- *Pick N samples* x_i *from proposal* $Q(X)$
- *If we knew* $w_i = P(x_i)/Q(x_i)$*, could do IS*
- *Instead, set* $\hat{w}_i = ZP(x_i)/Q(x_i)$

Parallel IS

$$\begin{aligned} E(\hat{W}) &= \int Q(x)(ZP(x)/Q(x))dx \\ &= Z\int P(x)dx \\ &= Z \end{aligned}$$

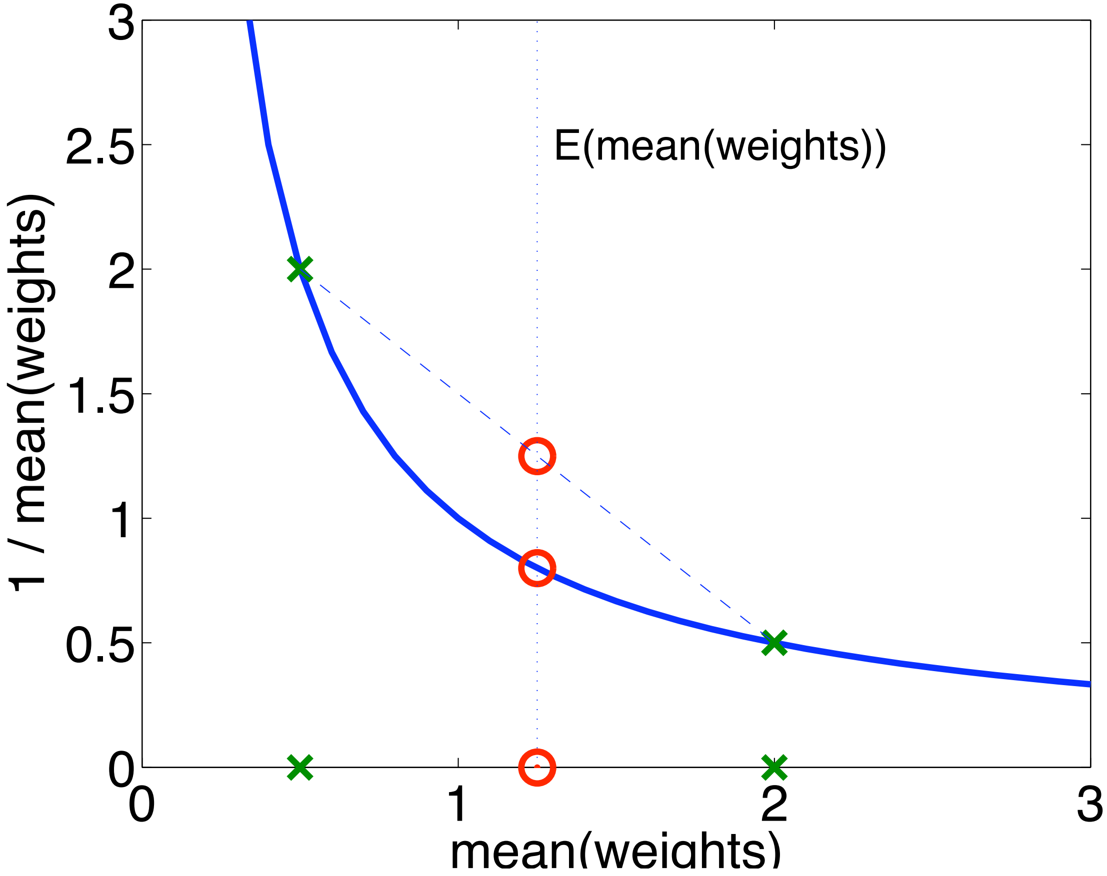
- $So, \bar{w} = \frac{1}{N}\sum_i \hat{w}_i \; is\ an\ unbiased\ estimate\ of\ Z$

Parallel IS

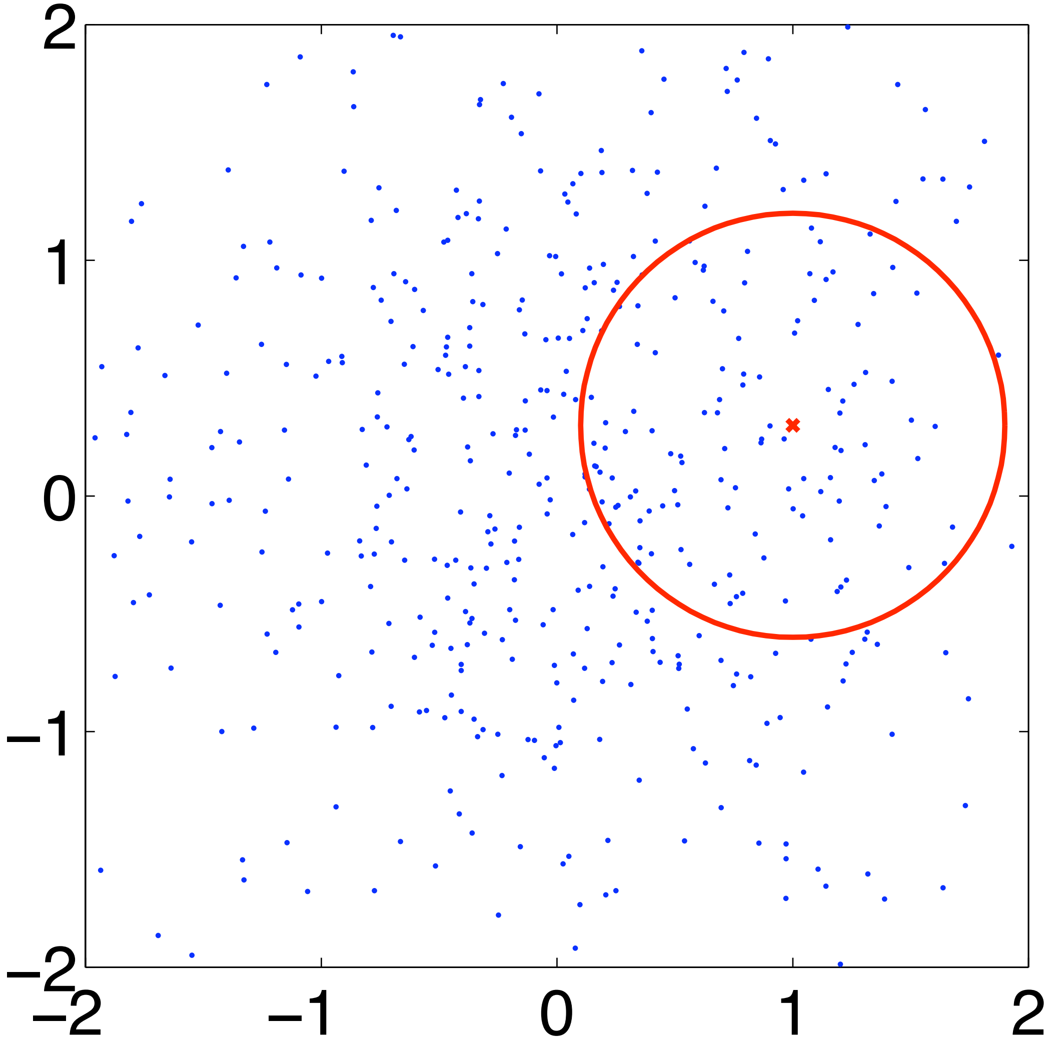
- *So,* $\hat{w}_i/\bar{w}$ *is an estimate of* w_i*, computed without knowing Z*
- *Final estimate:*

$$\int f(x)dx \approx \frac{1}{n}\sum_i \frac{\hat{w}_i}{\bar{w}} g(x_i)$$

Parallel IS is biased

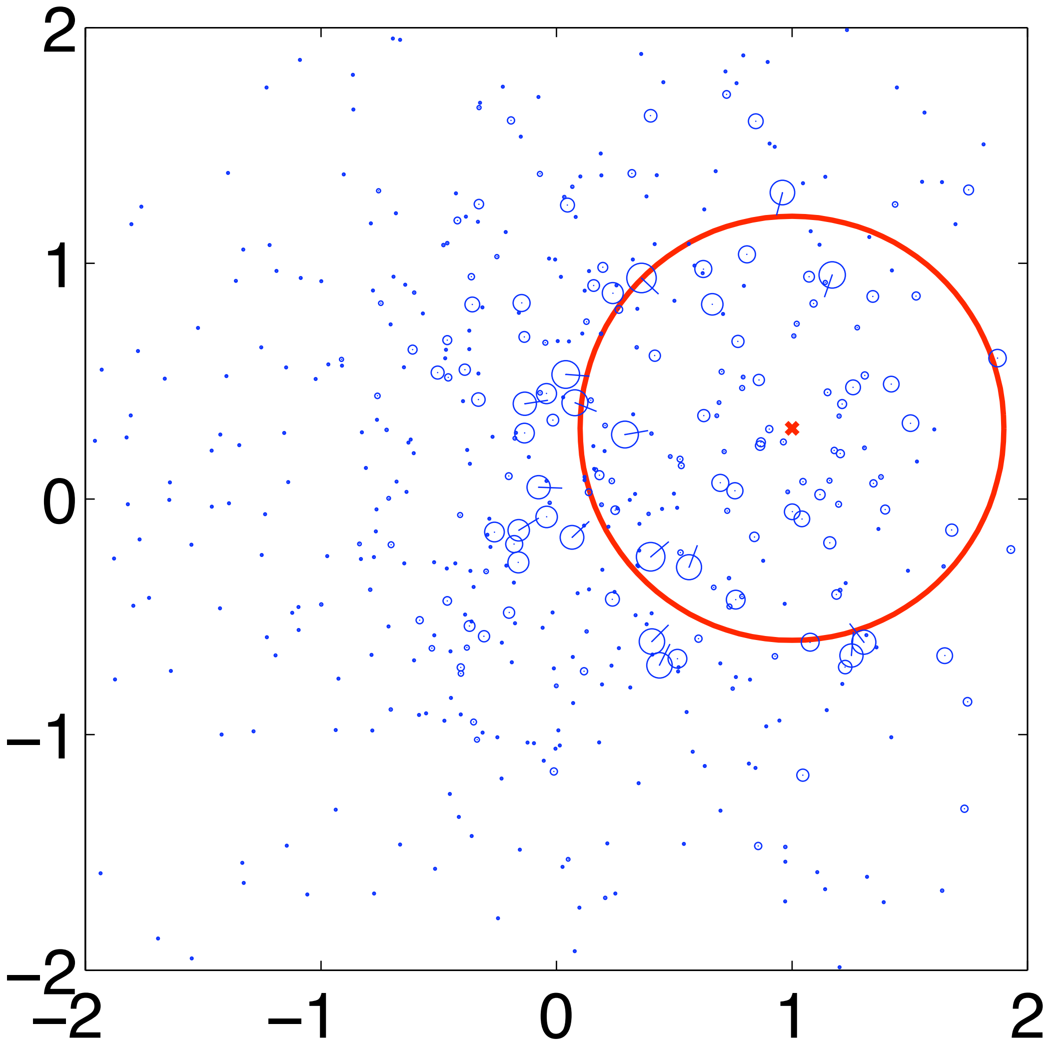


$E(\bar{W}) = Z$, but $E(1/\bar{W}) \neq 1/Z$ in general



$$Q : (X, Y) \sim N(1, 1) \qquad \theta \sim U(-\pi, \pi)$$

$$f(x, y, \theta) = Q(x, y, \theta) P(o = 0.8 \mid x, y, \theta)/Z$$



$$Posterior\ \ E(X, Y, \theta) = (0.496, 0.350, 0.084)$$

Back to high dimensions

- *Picking a good importance distribution is hard in high-D*
- *Major contributions to integral can be hidden in small areas*
  - *recall, want (f big ==> Q big)*
- *Would like to search for areas of high f(x)*
- *But searching could bias our estimates*

MCMC

Markov-Chain Monte Carlo

- *Design a randomized search procedure M which tends to increase f(x) if it is small*
- *Run M for a while, take resulting x as a sample*
- *Importance weight Q(x)?*

Markov-Chain Monte Carlo

- *Design a randomized search procedure M which tends to increase f(x) if it is small*
- *Run M for a while, take resulting x as a sample*
- *Importance weight Q(x)? Stationary distribution of M*

Designing a search chain

$$\int f(x)dx = \int P(x)g(x)dx = E_P(g(x))$$

- *Would like Q(x) = P(x)*
- *Turns out we can get this exactly, using Metropolis-Hastings*

Metropolis-Hastings

- *Way of designing chain w/* $Q(x) = P(x)$
- *Basic strategy: start from arbitrary x*
- *Repeatedly tweak x to get x'*
- *If* $P(x') \geq P(x)$*, move to x'*
- *If* $P(x') << P(x)$*, stay at x*
- *In intermediate cases, randomize*

Proposal distribution

- *Left open: what does "tweak" mean?*
- *Parameter of MH: Q(x' | x)*
  - *one-step proposal distribution*
- *Good proposals explore quickly, but remain in regions of high P(x)*
- *Optimal proposal?*

MH algorithm

- *Sample x' ~ Q(x' | x)*
- *Compute p =* $\frac{P(x')}{P(x)} \frac{Q(x \mid x')}{Q(x' \mid x)}$
- *With probability p, set x := x'*
- *Repeat for T steps (usually < T distinct samples)*

MH notes

- *Only need P(x) up to a constant factor*
- *Efficiency determined by:*
  - *how fast Q(x' | x) moves us around*
  - *how high acceptance probability p is*
- *Tension between fast Q and high p*