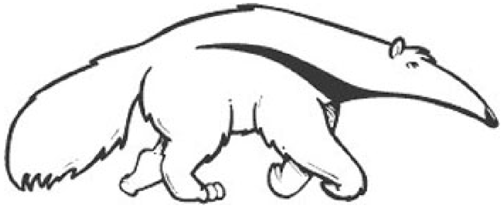
Algorithms for Reasoning with Probabilistic Graphical Models

Class 3: Approximate Inference

International Summer School on Deep Learning
July 2017

Prof. Rina Dechter
Prof. Alexander Ihler



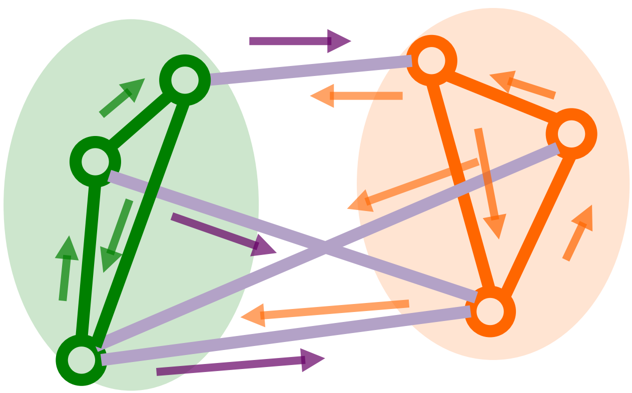
BREN:ICS INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

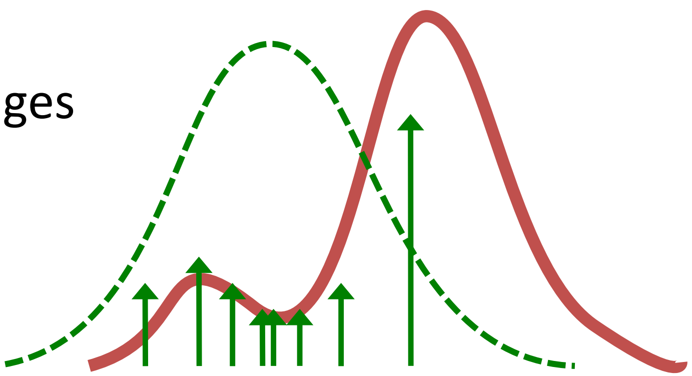
Approximate Inference

- Two main schools of approximate inference

- **Variational methods**
  - Frame "inference" as convex optimization
    & approximate (constraints, objectives)
  - Reason about "beliefs"; pass messages
  - Fast approximations & bounds
  - Quality often limited by memory



- **Monte Carlo sampling**
  - Approximate expectations with sample averages
  - Estimates are asymptotically correct
  - Can be hard to gauge finite sample quality



Graphical models

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Example:

$A \in \{0, 1\}$

$B \in \{0, 1\}$

$C \in \{0, 1\}$

$f_{AB}(A, B), \quad f_{BC}(B, C)$

The *combination operator* defines an overall function from the individual factors,

e.g., "+" : $F(A, B, C) = f_{AB}(A, B) + f_{BC}(B, C)$

Notation:

Discrete Xi) values called "states"

"Tuple" or "configuration": states taken by a set of variables

"Scope" of f: set of variables that are arguments to a factor f

often index factors by their scope, e.g., $f_\alpha(X_\alpha), \quad X_\alpha \subseteq X$

Graphical models

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Example:

$A \in \{0,1\}$

$B \in \{0,1\}$

$C \in \{0,1\}$

$f_{AB}(A,B), \quad f_{BC}(B,C)$

$$F(A,B,C) = f_{AB}(A,B) + f_{BC}(B,C)$$

For discrete variables, think of functions as "tables"
(though we might represent them more efficiently)

A	B	f(A,B)
0	0	6
0	1	0
1	0	0
1	1	6

+

B	C	f(B,C)
0	0	6
0	1	0
1	0	0
1	1	6

=

A	B	C	f(A,B,C)
0	0	0	12
0	0	1	6
0	1	0	0
0	1	1	6
1	0	0	6
1	0	1	0
1	1	0	6
1	1	1	12

$F(A=0, B=1, C=1)$ = 0 + 6

Canonical forms

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Typically either multiplication or summation; mostly equivalent:

$$f_\alpha(X_\alpha) \geq 0$$

$$F(X) = \prod_\alpha f_\alpha(X_\alpha)$$

log / exp

$$\theta_\alpha(X_\alpha) = \log f_\alpha(X_\alpha) \in \mathbb{R}$$

$$\theta(X) = \log F(x) = \sum_\alpha \theta_\alpha(X_\alpha)$$

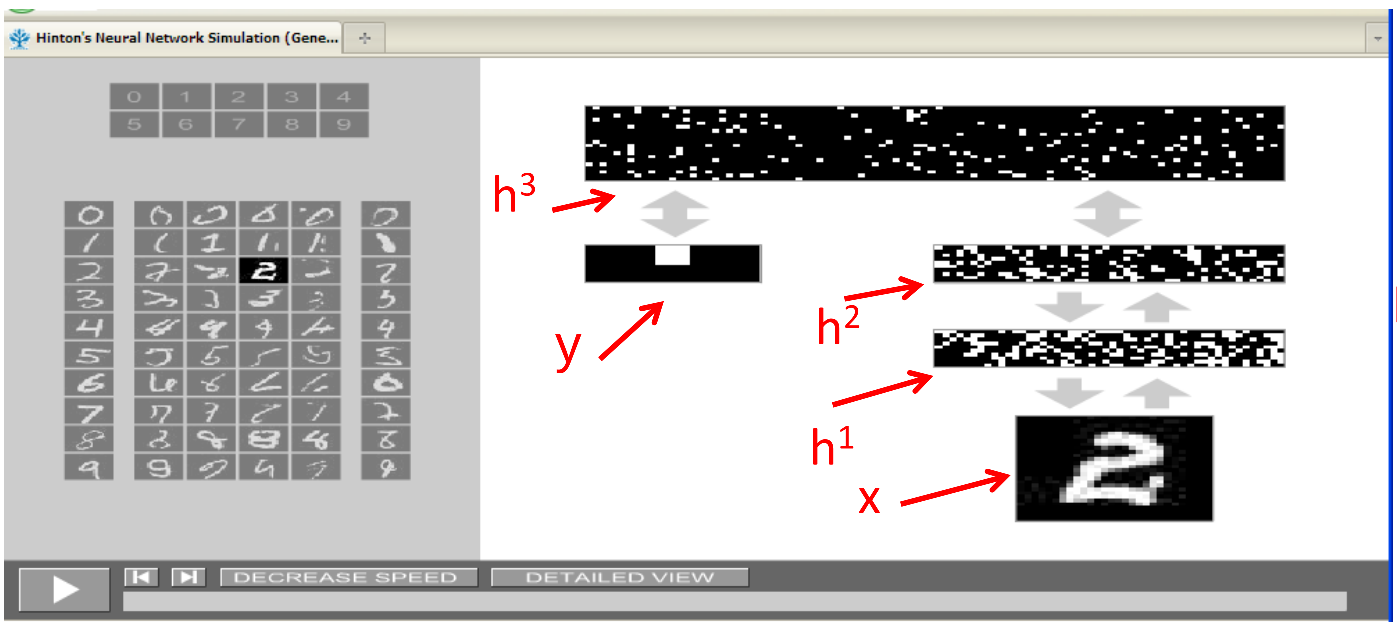
Product of nonnegative factors
(probabilities, 0/1, etc.)

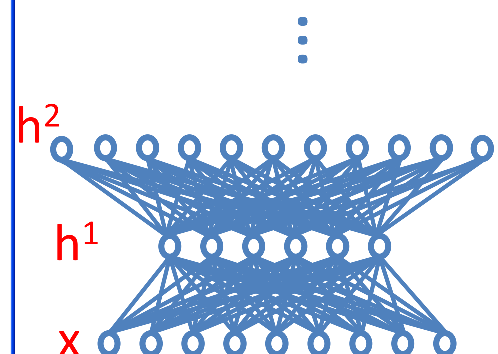
Sum of factors
(costs, utilities, etc.)

Ex: DBMs

[Hinton et al. 2007]

- Example: Deep Boltzmann machines
  - 784 pixels ⇔ 500 mid ⇔ 500 high ⇔ 2000 top ⇔ 10 labels
    x h^1 h^2 h^3 y
  - Induced width? ~2000!



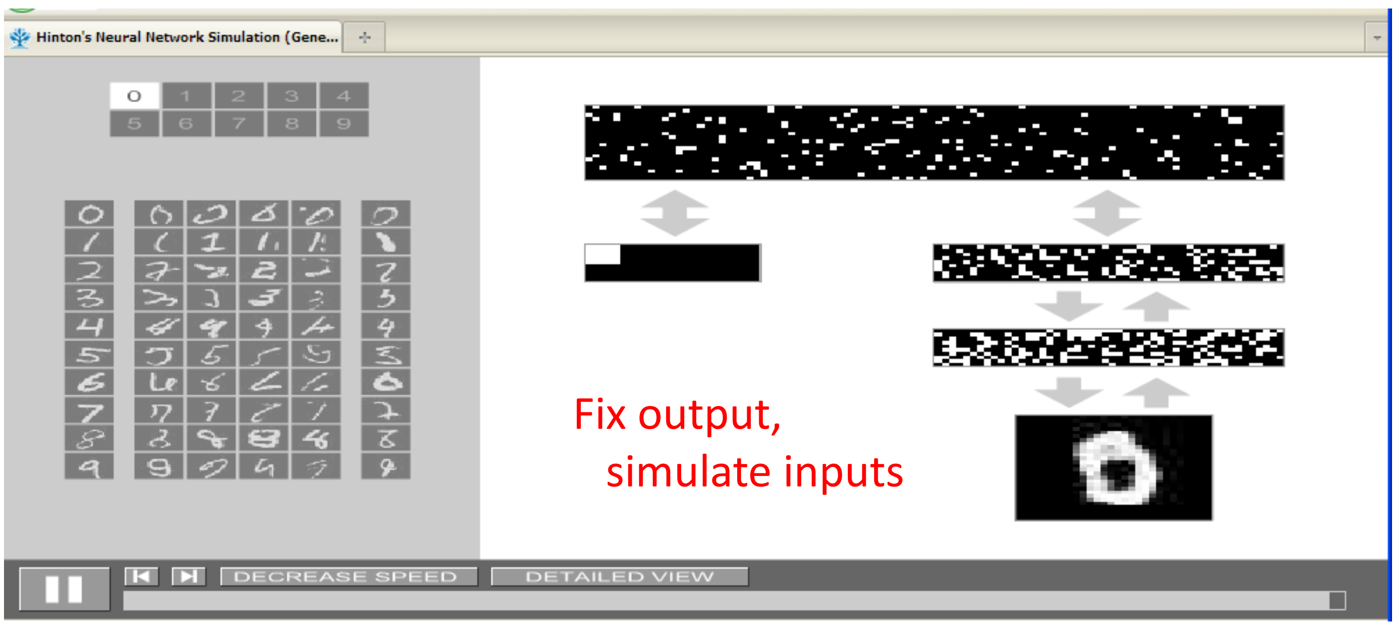


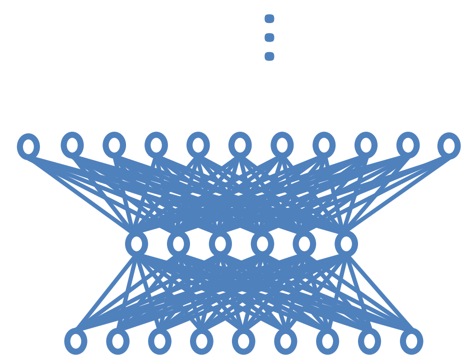
(¼ 1.5m parameters)

Ex: DBMs

[Hinton et al. 2007]

- Example: Deep Boltzmann machines
  - 784 pixels ⇔ 500 mid ⇔ 500 high ⇔ 2000 top ⇔ 10 labels
  - Induced width? ~2000!
  - Generative model: can simulate data, use partial observations, …

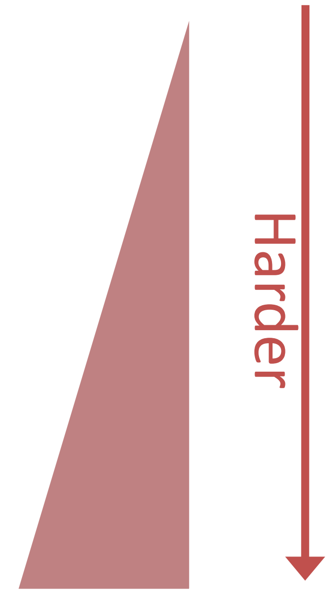




(¼ 1.5m parameters)

Types of queries

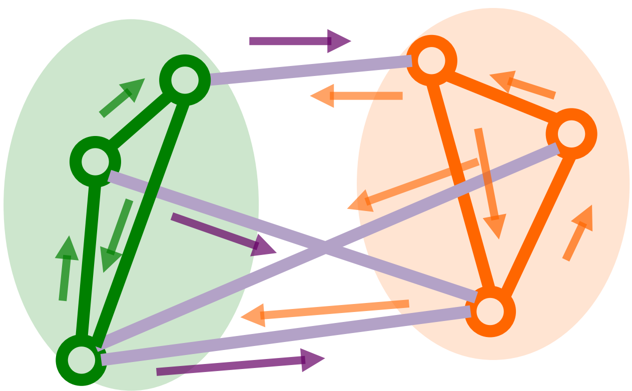
Max-Inference	$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Sum-Inference	$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Mixed-Inference	$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$



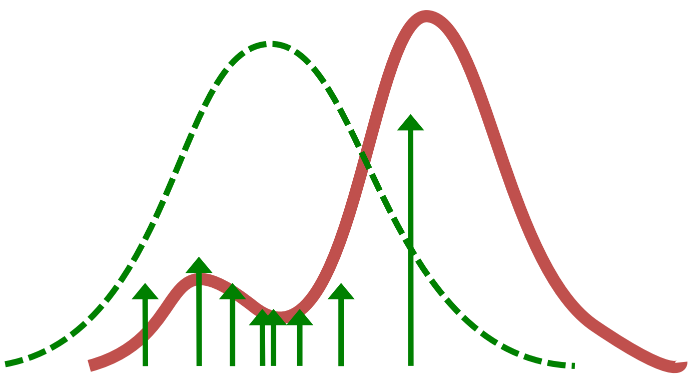
- **NP-hard**: exponentially many terms
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate ! Slower & more accurate

Outline

- Review: Graphical Models
- **Variational methods**
  - Convexity & decomposition bounds
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships



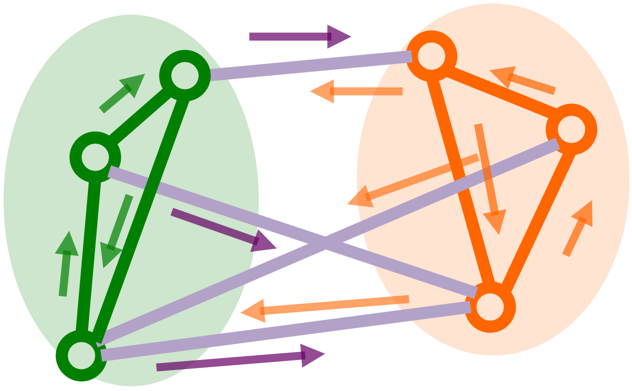
- Monte Carlo sampling
  - Basics
  - Importance sampling
  - Markov chain Monte Carlo
  - Integrating inference and sampling



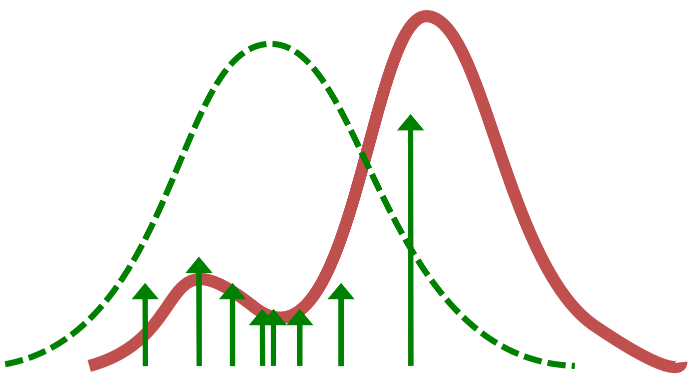
Outline

- Review: Graphical Models

- **Variational methods**
  - **Convexity & decomposition bounds**
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships



- Monte Carlo sampling
  - Basics
  - Importance sampling
  - Markov chain Monte Carlo
  - Integrating inference and sampling

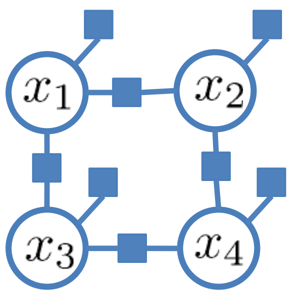


Vector space representation

- Represent the (log) model and state in a vector space

$$\theta(x) = \theta_1(x_1) + \theta_2(x_2) + \ldots + \theta_{12}(x_1, x_2) + \ldots$$

$$x = [x_1, x_2, x_3, x_4] = [\,1\;,\;0\;,\;1\;,\;1\,]$$



X_1	$\theta(X_1)$
0	0.5
1	0.75

…

X_1	X_2	$\theta(X_1, X_2)$
0	0	-1.0
0	1	2.0
1	0	0.0
1	1	1.0

…

$$\vec{\theta} = \begin{bmatrix} 0.5 \\ 0.75 \\ . \\ . \\ \ldots \\ -1.0 \\ 2.0 \\ 0.0 \\ 1.0 \\ \ldots \end{bmatrix} \qquad \begin{array}{l} \theta_1(x_1) \\ \theta_2(x_2) \\ \theta_{12}(x_1, x_2) \end{array}$$

$$\vec{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \ldots \\ 0 \\ 0 \\ 1 \\ 0 \\ \ldots. \end{bmatrix} \qquad \begin{array}{l} x_1 = 1 \\ x_2 = 0 \\ (x_1, x_2) = (1, 0) \end{array}$$

Evaluating the function is a dot product in the vector space:

$$\theta(x) \;=\; \vec{\theta} \cdot \vec{x}$$

Inference Tasks & Convexity

- Distribution is log-linear (exponential family):

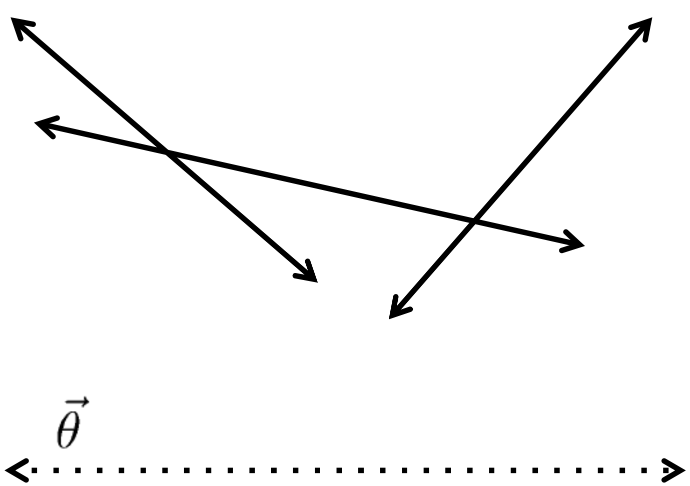
$$p(x) = \frac{1}{Z} f(x) \propto \exp\left[\vec{\theta} \cdot u(x)\right]$$

$\vec{\theta}$ "natural parameters"

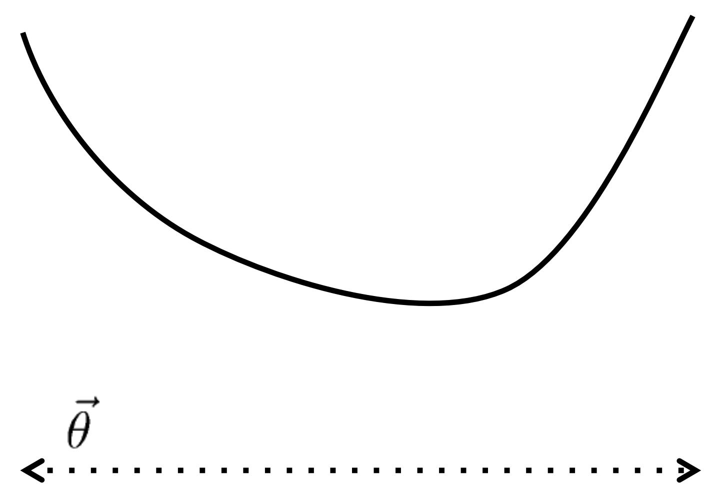
$u(x) = \vec{x}$ "features"

- Tasks of interest are convex functions of the model:

$$\log f(\mathbf{x}^*) = \max_{\vec{x} \in \mathcal{X}} \vec{\theta} \cdot \vec{x}$$



$$\log Z = \log \sum_{\vec{x} \in \mathcal{X}} \exp\left[\vec{\theta} \cdot \vec{x}\right]$$

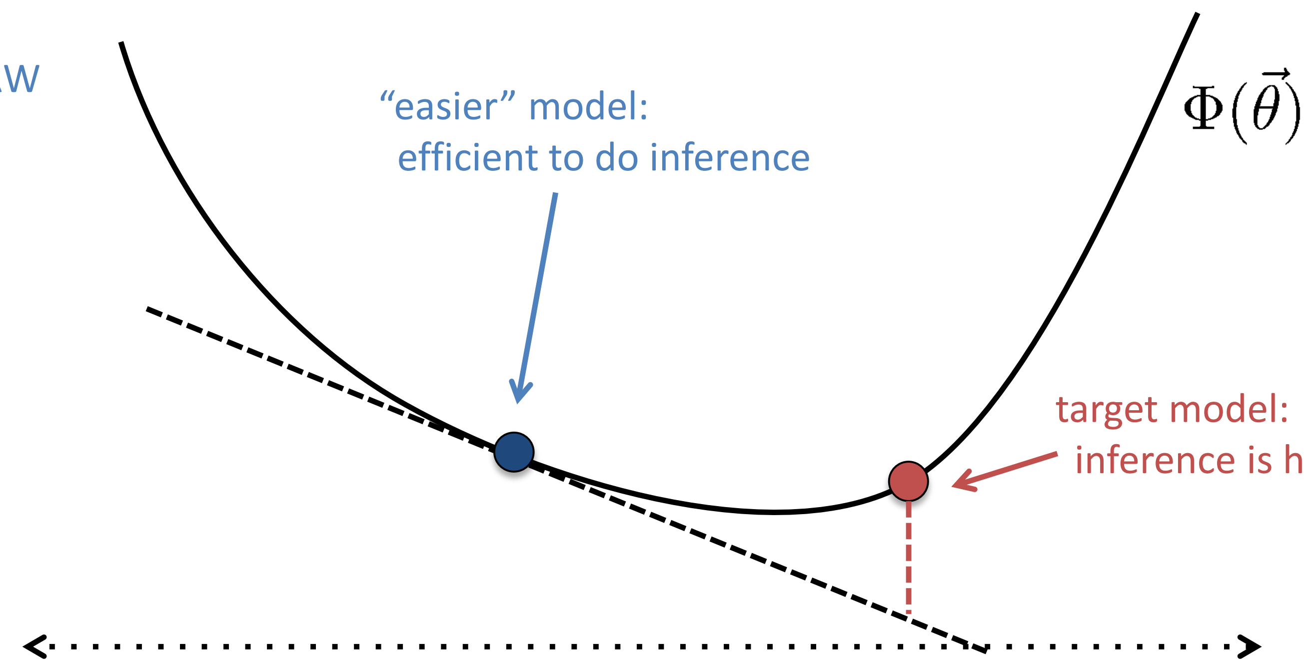


Bounds via Convexity

- Convexity relates target to "nearby" models
  - Some of these models are easy to solve! (trees, etc.)
  - Inference at easy models + convexity tells us something about our model!

- Lower bounds:

Mean field
Negative TRW
…

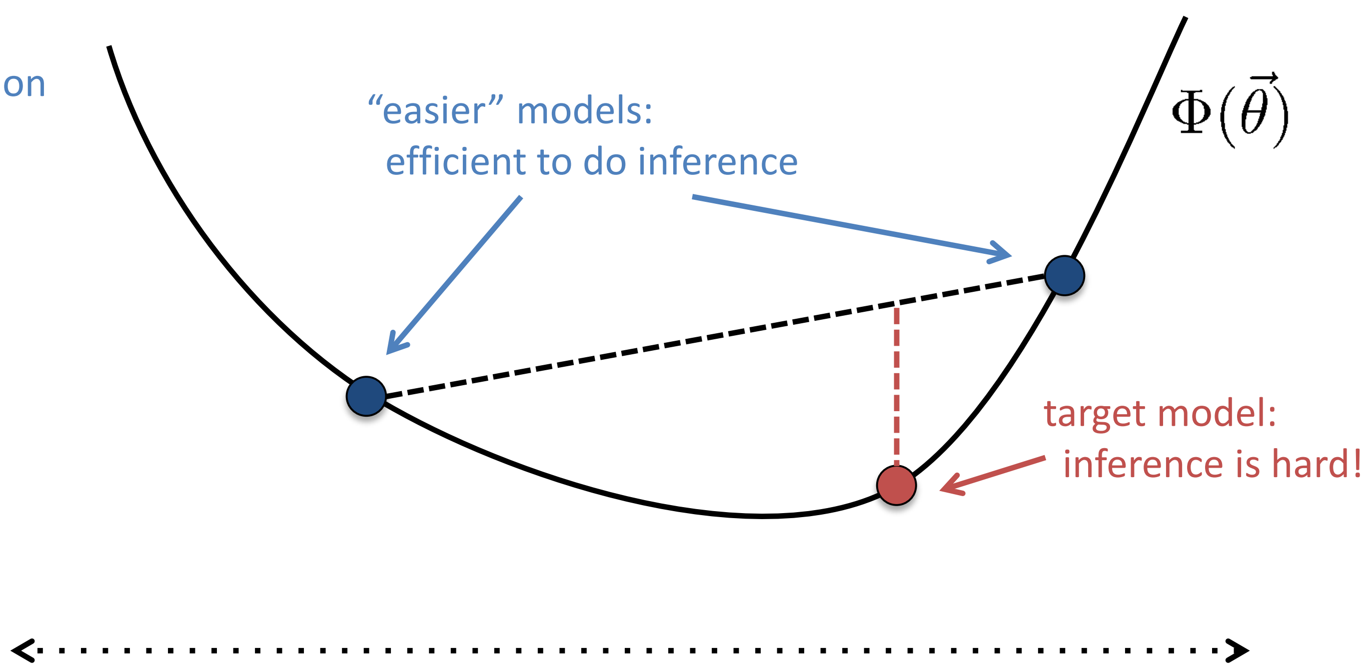


Bounds via Convexity

- Convexity relates target to "nearby" models
  - Some of these models are easy to solve! (trees, etc.)
  - Inference at easy models + convexity tells us something about our model!

- Upper bounds:

TRW
Decomposition
…



Tree-reweighted MAP

$$\Phi_0(\vec{\theta}) = \max_{\vec{x} \in \mathcal{X}} \vec{\theta} \cdot \vec{x}$$

- Let T_1, T_2 be two (or more) tree-structured models, with

$$\vec{\theta} \;=\; w_1\, \vec{\theta}^{(1)} + w_2\, \vec{\theta}^{(2)}$$

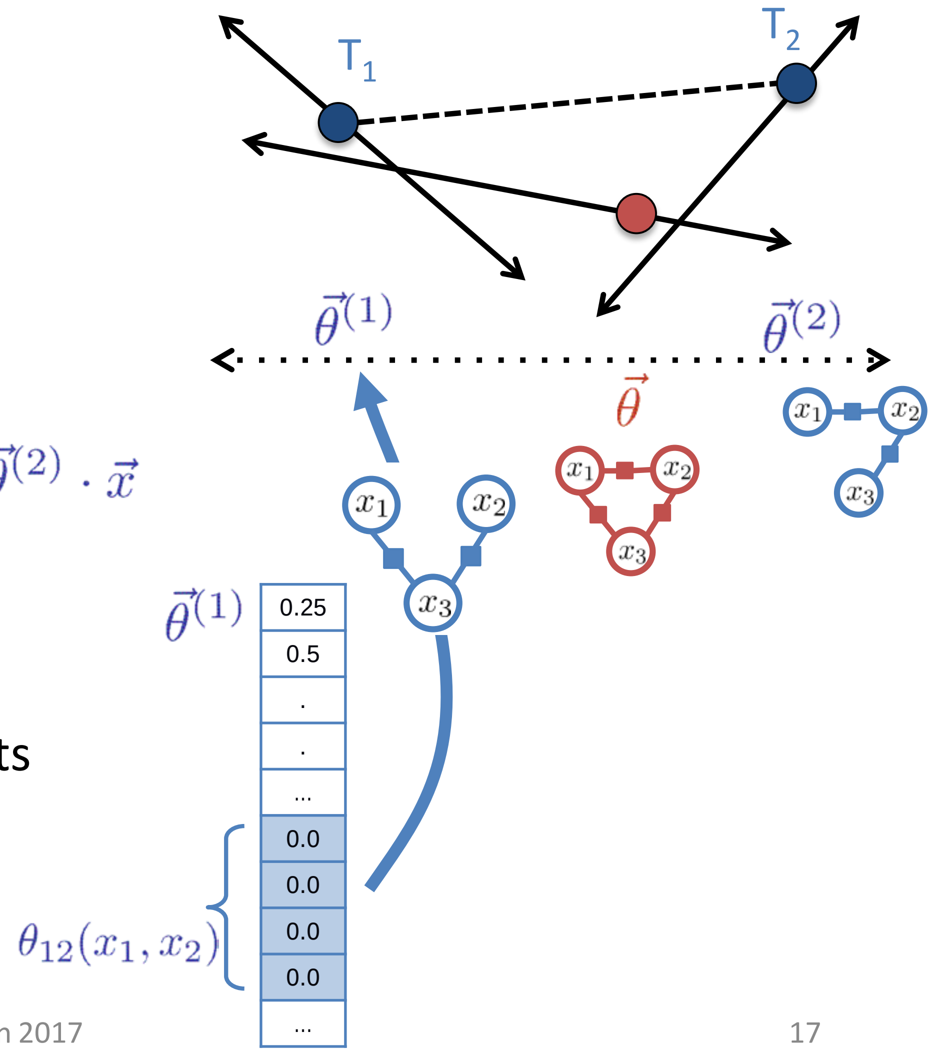
- Each T_i is easy to solve:

$$\vec{x}^{*(1)} = \max_{\vec{x}}\ \vec{\theta}^{(1)} \cdot \vec{x}$$

- And by convexity,

$$\max_{\vec{x}} \vec{\theta} \cdot \vec{x} \;\le\; w_1 \max_{\vec{x}} \vec{\theta}^{(1)} \cdot \vec{x} + w_2 \max_{\vec{x}} \vec{\theta}^{(2)} \cdot \vec{x}$$

- Minimize bound?
  – Convex objective, linear constraints



Decomposition Bounds

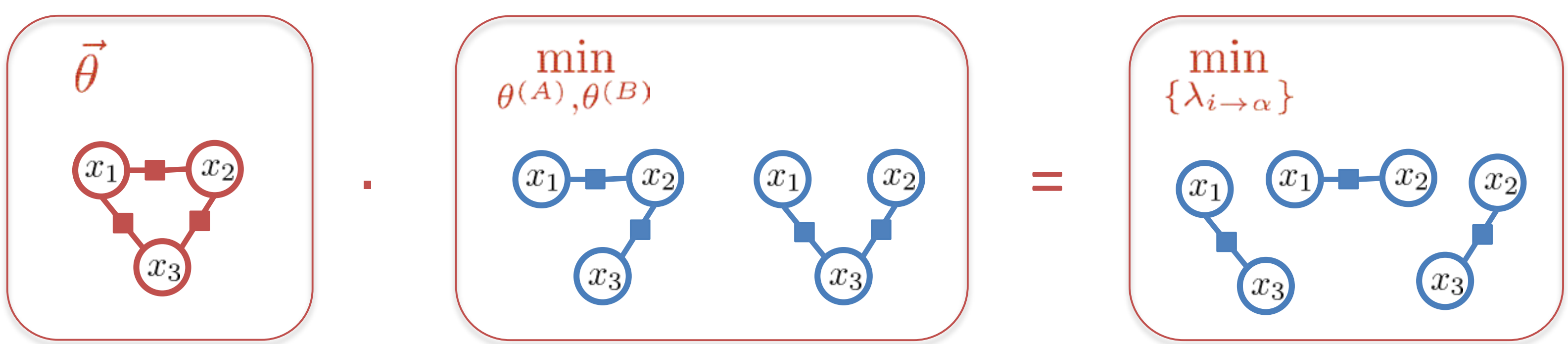
- TRW MAP is equivalent to MAP decomposition

$$\max_{\vec{x}} \left[\vec{\theta} \cdot \vec{x}\right] \le \min_{\theta^{(1)},\theta^{(2)}} \max_{\vec{x}} \left[w_1 \vec{\theta}^{(1)} \cdot \vec{x}\right] + \max_{\vec{x}} \left[w_2 \vec{\theta}^{(2)} \cdot \vec{x}\right] \qquad \vec{\theta} = w_1 \vec{\theta}^{(1)} + w_2 \vec{\theta}^{(2)}$$

$$= \min_{\theta^{(A)},\theta^{(B)}} \max_{\vec{x}} \left[\vec{\theta}^{(A)} \cdot \vec{x}\right] + \max_{\vec{x}} \left[\vec{\theta}^{(B)} \cdot \vec{x}\right] \qquad \vec{\theta} = \vec{\theta}^{(A)} + \vec{\theta}^{(B)}$$

$$= \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\vec{x}_\alpha} \left[(\vec{\theta}_\alpha + \sum_i \vec{\lambda}_{i\to\alpha}) \cdot \vec{x}_\alpha\right] \qquad \vec{0} = \sum_{\alpha \ni i} \vec{\lambda}_{i\to\alpha}$$

(on trees, decomposition bound = exact inference)



More compact
Faster optimization
Reparameterization "messages"

Tree-reweighted Sum

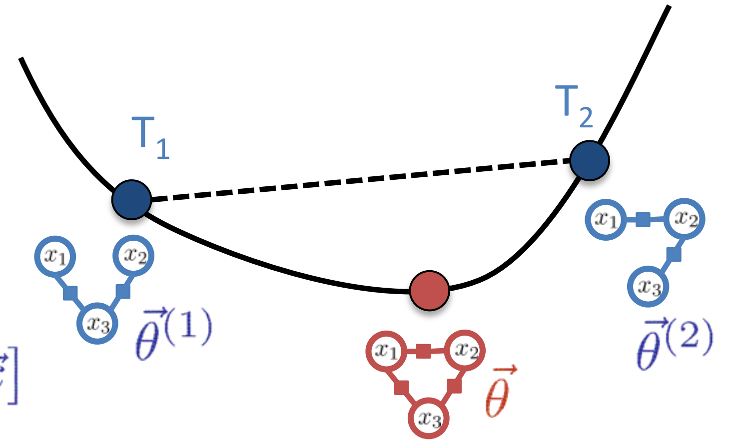
$$\Phi_1(\vec{\theta}) = \log \sum_{\vec{x} \in \mathcal{X}} \exp\left[\vec{\theta} \cdot \vec{x}\right]$$

- Let T_1, T_2 be two (or more) tree-structured models, with

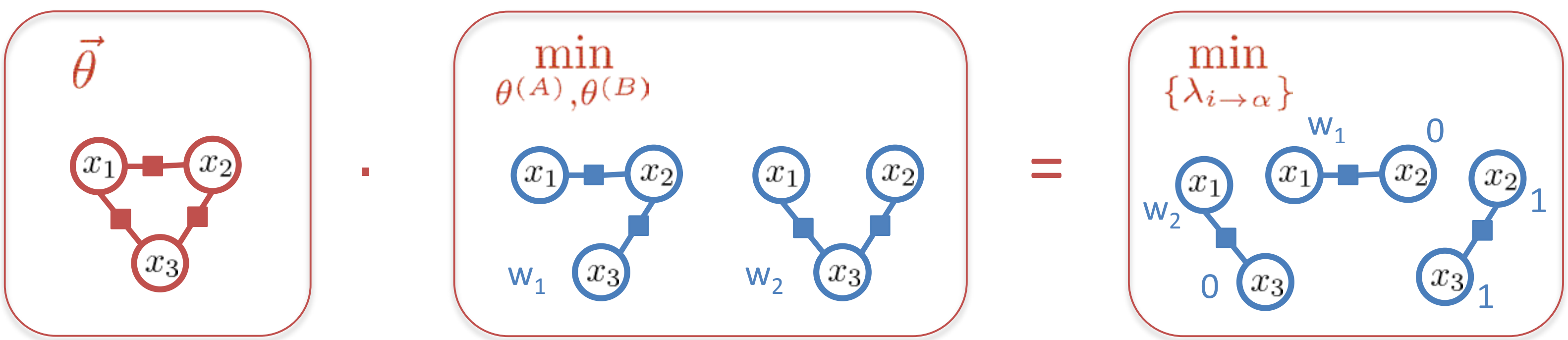
$$\vec{\theta} \;=\; w_1\, \vec{\theta}^{(1)} + w_2\, \vec{\theta}^{(2)} \;=\; \vec{\theta}^{(A)} + \vec{\theta}^{(B)}$$

- Again, we have

$$\Phi_1(\vec{\theta}) \;\leq\; w_1\, \Phi_1(\vec{\theta}^{(1)}) + w_2\, \Phi_1(\vec{\theta}^{(2)})$$
$$= \log \sum_{\vec{x}}^{w_1} \exp\left[\vec{\theta}^{(A)} \vec{x}\right] + \log \sum_{\vec{x}}^{w_2} \exp\left[\vec{\theta}^{(B)} \vec{x}\right]$$



$$\sum_{x}^{w} f(x) = \left[\sum_{x} f(x)^{\frac{1}{w}}\right]^{w}$$



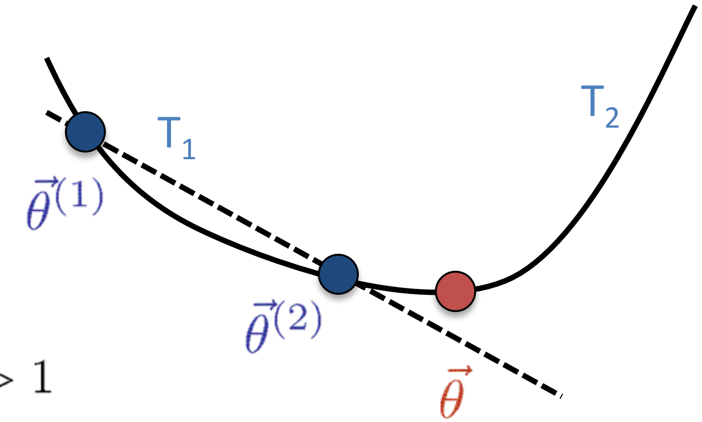
(if T_1, T_2 share an elimination order)

Negative TRW

$$\Phi_1(\vec{\theta}) = \log \sum_{\vec{x} \in \mathcal{X}} \exp\left[\vec{\theta} \cdot \vec{x}\right]$$

- We can also get a lower bound via decomposition:

$$\begin{aligned} \vec{\theta} &= \vec{\theta}^{(1)} + \alpha\,(\vec{\theta}^{(2)} - \vec{\theta}^{(1)}) \\ &= w_1\,\vec{\theta}^{(1)} + w_2\,\vec{\theta}^{(2)} = \vec{\theta}^{(A)} + \vec{\theta}^{(B)} \end{aligned}$$



$\alpha > 1$

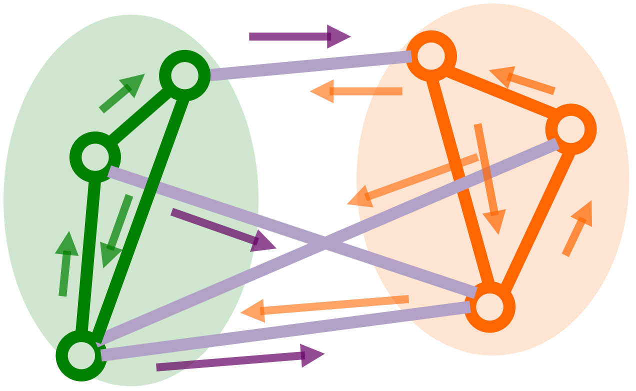
$\Rightarrow w_2 > 1,\ w_1 < 0$

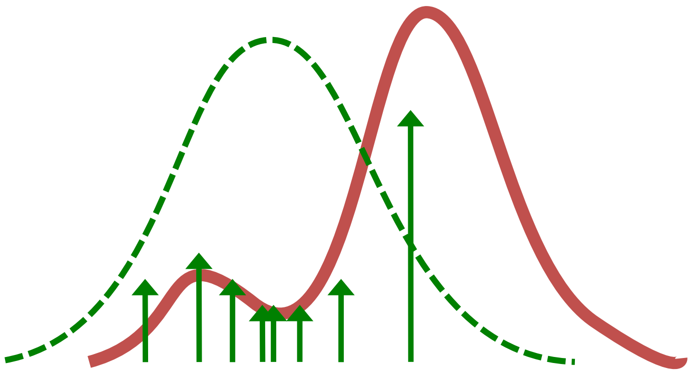
- Identical bound computation,
  but with all weights but one negative:

$$\begin{aligned} \Phi_1(\vec{\theta}) &\geq w_1\,\Phi_1(\vec{\theta}^{(1)}) + w_2\,\Phi_1(\vec{\theta}^{(2)}) \\ &= \log \sum_{\vec{x}}^{w_1} \exp\left[\vec{\theta}^{(A)}\vec{x}\right] + \log \sum_{\vec{x}}^{w_2} \exp\left[\vec{\theta}^{(B)}\vec{x}\right] \end{aligned}$$

Outline

- Review: Graphical Models
- **Variational methods**
  - Convexity & decomposition bounds
  - **Variational forms & the marginal polytope**
  - Message passing algorithms
  - Convex duality relationships
- Monte Carlo sampling
  - Basics
  - Importance sampling
  - Markov chain Monte Carlo
  - Integrating inference and sampling





Variational forms

- Reframe inference task as an optimization over distributions q(x)
- Ex: MAP inference $\max_x \log f(x) = \log f(x^*) = \max_{q \in \mathbb{P}} \mathbb{E}_q[\log f(x)]$

  Optimal q(x) puts all mass on optimal value(s) of x: $q^*(x) = \mathbb{1}[x = x^*]$
  (mass on any other values of x reduces the average)

- Sum inference: $\log Z = \log \sum_x f(x) = \max_{q \in \mathbb{P}} \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$

  Proof:

$$D(q\|p) = \sum_x q(x) \log \left[\frac{q(x)}{\frac{1}{Z} f(x)} \right]$$

(Kullback–Leibler divergence)

$$= -H(x\,;\,q) - \mathbb{E}_q[\log f(x)] + \log Z$$

$$\Rightarrow \log Z \geq \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

Equal iff
$q(x) = p(x) = \frac{1}{Z} f(x)$

- How to optimize over distributions q?

The marginal polytope

- Rewrite $\log f(x^*) = \max_{q \in \mathbb{P}} \mathbb{E}_q[\log f(x)] = \max_{q \in \mathbb{P}} \mathbb{E}_q[\vec{\theta} \cdot \vec{x}] \quad = \quad \max_{\vec{\mu} \in \mathcal{M}} \vec{\theta} \cdot \vec{\mu}$

and similarly, $\log Z \;=\; \max_{\vec{\mu} \in \mathcal{M}} \vec{\theta} \cdot \vec{\mu} + H(\vec{\mu})$

(max entropy given [1])

$\vec{\mu} = \mathbb{E}_q[\vec{x}]$

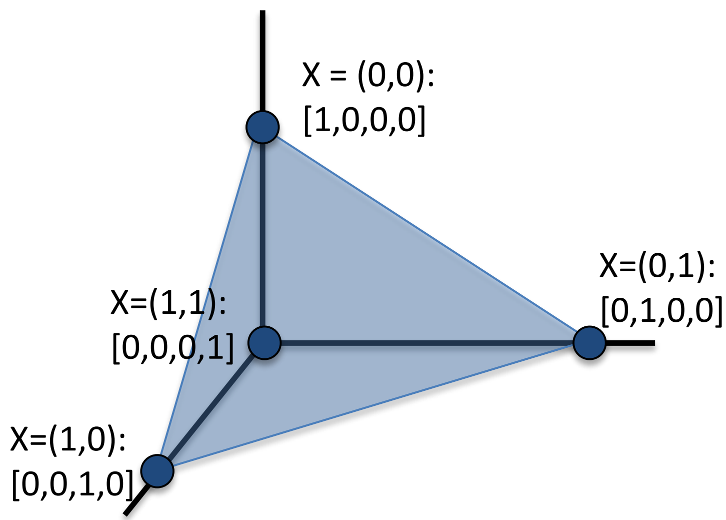
(the marginal probabilities of q)

$\vec{x}$		$\vec{\mu} = \mathbb{E}_q[\vec{x}]$
0	$x_1 = 1$	q(X_1=0)
1		q(X_1=1)
1	$x_2 = 0$	q(X_2=0)
0		.
0	$(x_1, x_2) = (1, 0)$	q(X_1=0,X_2=0)
0		q(X_1=0,X_2=1)
1		q(X_1=1,X_2=0)
0		.

$\mathcal{M} = \{\, \vec{\mu} \;:\; \exists q : \vec{\mu} = \mathbb{E}_q[\vec{x}] \,\}$

(set of all valid marginal probabilities of q)

"marginal polytope"



Variational perspectives

- Replace q 2 P and H(q) with simpler approximations

$$\log p(x^*) = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)]$$

$$\log Z = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

- Algorithms and their properties:

	Method	distributions	entropy	value
Max:	Linear programming	$q \in \mathbb{L} \supseteq \mathbb{P}$	n/a	$\hat{p}_{lp} \geq p(x^*)$
Sum:	Mean field	$\{q = \prod q_i(x_i)\} \subseteq \mathbb{P}$	exact	$Z_{mf} \leq Z$
	Belief propagation	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_\beta \approx H(q)$	$Z_\beta \approx Z$
	Tree-reweighted	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_{tr} \geq H(q)$	$Z_{tr} \geq Z$

Variational perspectives

- Replace q 2 P and H(q) with simpler approximations

$$\log p(x^*) = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)]$$

$$\log Z = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

- Algorithms and their properties:

	Method	distributions	entropy	value
Max:	Linear programming	$q \in \mathbb{L} \supseteq \mathbb{P}$	n/a	$\hat{p}_{lp} \geq p(x^*)$
Sum:	**Mean field**	$\{q = \prod q_i(x_i)\} \subseteq \mathbb{P}$	exact	$Z_{mf} \leq Z$
	Belief propagation	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_\beta \approx H(q)$	$Z_\beta \approx Z$
	Tree-reweighted	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_{tr} \geq H(q)$	$Z_{tr} \geq Z$

Mean Field

- We can design lower bounds by restricting q(x)
  - Naïve mean field: q(x) is fully independent
  - Entropy H(q) is then easy:

$$q(x) = \prod_i q_i(x_i)$$

) $$H(q) = \sum_i H(q_i)$$

- Optimizing the bound via coordinate ascent:

$$\begin{aligned}\mathbb{E}_q[\theta(x)] + H(q) &= \mathbb{E}_q\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big] + H(q_i) + \text{const} \\ &= \mathbb{E}_{q_i}\big[\log g(x_i)\big] + H(q_i) \\ &= D(\, q_i \,\|\, g_i \,)\end{aligned}$$

$$\log g_i(x_i) = \mathbb{E}_{q_{\neg i}}\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big]$$

$$q_{\neg i}(x) = \prod_{j \neq i} q_j(x_j)$$

Coordinate update:

) $$q_i(x_i) \propto \exp\Big[\, \mathbb{E}_{q_{\neg i}}\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big]\Big]$$

Mean Field

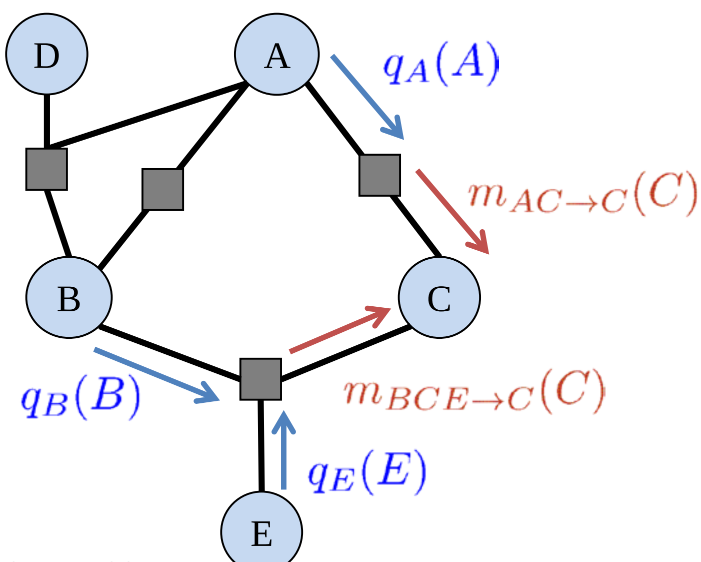
- We can design lower bounds by restricting q(x)
  - Naïve mean field: q(x) is fully independent
  - Entropy H(q) is then easy:

$$q(x) = \prod_i q_i(x_i)$$

$$) \quad H(q) = \sum_i H(q_i)$$

- Optimizing the bound via coordinate ascent:

$$q_i(x_i) \propto \exp\Big[\, \mathbb{E}_{q_{\neg i}}\big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\big]\Big]$$



"Message passing" interpretation:
Updates depend only on Xi's Markov blanket

Naïve Mean Field

Initialize $\{q_i(X_i)\}$
while not converged **do**
  for $i = 1 \ldots n$ **do**
    $m_{\alpha\to i}(x_i) = \exp\Big[\sum_{x_{\alpha\setminus i}} \theta_\alpha(x_\alpha) \prod_{j\in\alpha\setminus i} q_j(x_j)\Big]$
    $q_i(x_i) \propto \prod_{\alpha\ni i} m_{\alpha\to i}(x_i)$

Naïve Mean Field

- Subset of M corresponding to independent distributions?
  - Includes all vertices (configurations of x), but not all distributions
  - Non-convex set; coordinate ascent has local optima

$$q(x) = \prod_i q_i(x_i)$$

$\vec{\mu} = \mathbb{E}_q[\vec{x}]$

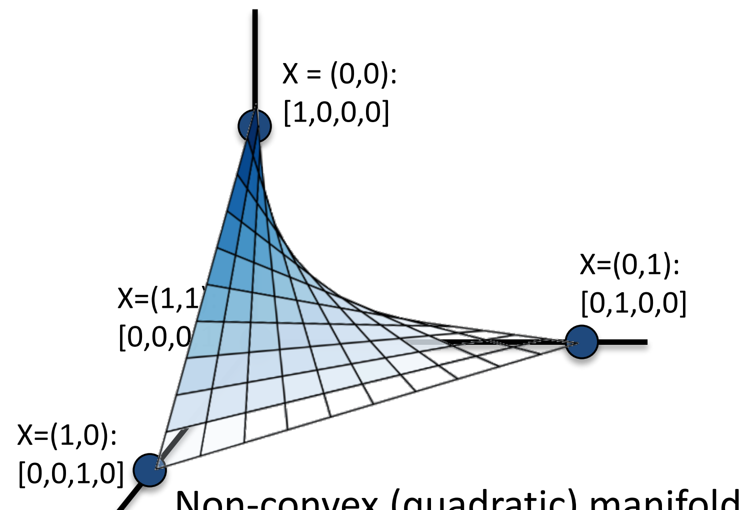
$q(X_1=0)$
$q(X_1=1)$
$q(X_2=0)$
$q(X_2=1)$
$q(X_1=0,X_2=0)$
$q(X_1=0,X_2=1)$
$q(X_1=1,X_2=0)$
.

$\vec{\mu} = \mathbb{E}_q[\vec{x}]$

$1-q_1$
q_1
$1-q_2$
q_2
$(1-q_1) \times (1-q_2)$
$(1-q_1) \times q_2$
$q_1 \times (1-q_2)$
.

$$\mathrm{MF} = \{\, \vec{\mu} \;:\; \exists\{q_i\} : \vec{\mu} = \mathbb{E}_q[\vec{x}] \,\}$$

(set of marginal probabilities of independent q)



Non-convex (quadratic) manifold!

Variational perspectives

- Replace q 2 P and H(q) with simpler approximations

$$\log p(x^*) = \max_{q \in \mathbb{P}} \ \mathbb{E}_q[\log f(x)]$$

$$\log Z = \max_{q \in \mathbb{P}} \ \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

- Algorithms and their properties:

	Method	distributions	entropy	value
Max:	**Linear programming**	$q \in \mathbb{L} \supseteq \mathbb{P}$	n/a	$\hat{p}_{lp} \geq p(x^*)$
Sum:	Mean field	$\{q = \prod q_i(x_i)\} \subseteq \mathbb{P}$	exact	$Z_{mf} \leq Z$
	Belief propagation	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_\beta \approx H(q)$	$Z_\beta \approx Z$
	Tree-reweighted	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_{tr} \geq H(q)$	$Z_{tr} \geq Z$

The local polytope

- Unfortunately, M has a large number of constraints
  - Enforce only a few, easy to check constraints?
  - Equivalent to a linear programming relaxation of original ILP

$\mu \in \mathbb{L}$: "local consistency" polytope

$\vec{\mu} = \mathbb{E}_q[\vec{x}]$

q(X_1=0)
q(X_1=1)
q(X_2=0)
q(X_2=1)
…
q(X_1=0,X_2=0)
q(X_1=0,X_2=1)
q(X_1=1,X_2=0)
q(X_1=1,X_2=0)
…

$$\left.\begin{array}{l} \mu_{i;k} \in [0,1] \\ \mu_{ij;kl} \in [0,1] \end{array}\right\}$$

All probabilities are within [0,1]

The local polytope

- Unfortunately, M has a large number of constraints
  - Enforce only a few, easy to check constraints?
  - Equivalent to a linear programming relaxation of original ILP

$\mu \in \mathbb{L}$: "local consistency" polytope

$$\vec{\mu} = \mathbb{E}_q[\vec{x}]$$

$q(X_1=0)$
$q(X_1=1)$
$q(X_2=0)$
$q(X_2=1)$
…
$q(X_1=0,X_2=0)$
$q(X_1=0,X_2=1)$
$q(X_1=1,X_2=0)$
$q(X_1=1,X_2=0)$
…

$$\mu_{i;k} \in [0,1]$$
$$\mu_{ij;kl} \in [0,1]$$

All probabilities are within [0,1]

$$\sum_k \mu_{i;k} = 1$$
$$\sum_{k,l} \mu_{ij;kl} = 1$$

Each marginal probability is normalized to sum to one

The local polytope

- Unfortunately, M has a large number of constraints
  - Enforce only a few, easy to check constraints?
  - Equivalent to a linear programming relaxation of original ILP

$\mu \in \mathbb{L}$: "local consistency" polytope

$$\vec{\mu} = \mathbb{E}_q[\vec{x}]$$

$q(X_1=0)$
$q(X_1=1)$
$q(X_2=0)$
$q(X_2=1)$
…
$q(X_1=0,X_2=0)$
$q(X_1=0,X_2=1)$
$q(X_1=1,X_2=0)$
$q(X_1=1,X_2=0)$
…

$$\mu_{i;k} \in [0,1]$$
$$\mu_{ij;kl} \in [0,1]$$

All probabilities are within [0,1]

$$\sum_k \mu_{i;k} = 1$$
$$\sum_{k,l} \mu_{ij;kl} = 1$$

Each marginal probability is normalized to sum to one

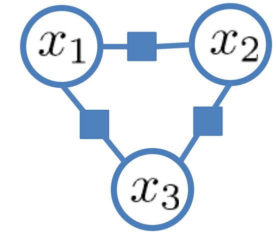
$$\sum_l \mu_{ij;kl} = \mu_{i;k}$$

Marginal of (x_i, x_j) is consistent with marginal of x_i

$$\sum_k \mu_{ij;kl} = \mu_{j;l}$$

(& similarly, consistent with x_j)

The local polytope



- Local polytope does not enforce all the constraints of M:
  - Ex: all pairwise probabilities locally consistent, but no joint q(x) exists:

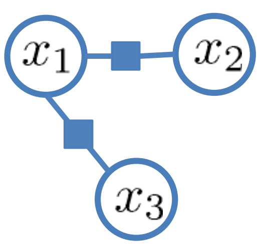
$$\mu_1 = \mu_2 = \mu_3 \quad \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

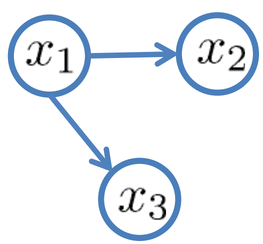
$$\mu_{12}: \quad \begin{array}{cc} & x_2 \\ x_1 & \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \end{array} \qquad \mu_{13}: \quad \begin{array}{cc} & x_3 \\ x_1 & \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \end{array} \qquad \mu_{23}: \quad \begin{array}{cc} & x_3 \\ x_2 & \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix} \end{array}$$

$$(x_1 = x_2) \qquad (x_1 = x_3) \qquad (x_2 \neq x_3)$$

(also illustrates connection to arc consistency in CSPs, etc.)

- But, trees remain easy
  - If we only specify the marginals on a tree, we can construct q(x)





$$q(x) = q(x_1) \cdot q(x_2|x_1) \cdot q(x_3|x_1)$$

$$= \mu_1 \cdot \frac{\mu_{12}}{\mu_1} \cdot \frac{\mu_{13}}{\mu_1}$$

$\mathbb{L} = \mathbb{M}$ on tree-structured distributions

Duality relationship

- Local polytope LP & MAP decomposition are Lagrangian duals:

$$\log f(x^*) \le \max_{\mu} \Big[\sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} \Big]$$

subject to (a) normalization constraints (enforce explicitly)

(b) consistency: $\sum_l \mu_{ij;kl} = \mu_{i;k}$, $\sum_k \mu_{ij;kl} = \mu_{j;l}$ (use Lagrange)

$$L = \max_{\mu} \min_{\lambda} \sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} + \sum_{i,j,k} \lambda_{i\to ij;k} \Big(\sum_l \mu_{ij;kl} - \mu_{i;k}\Big)$$

$$\le \min_{\lambda} \max_{\mu} \sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} + \sum_{i,j,k} \lambda_{i\to ij;k} \Big(\sum_l \mu_{ij;kl} - \mu_{i;k}\Big)$$

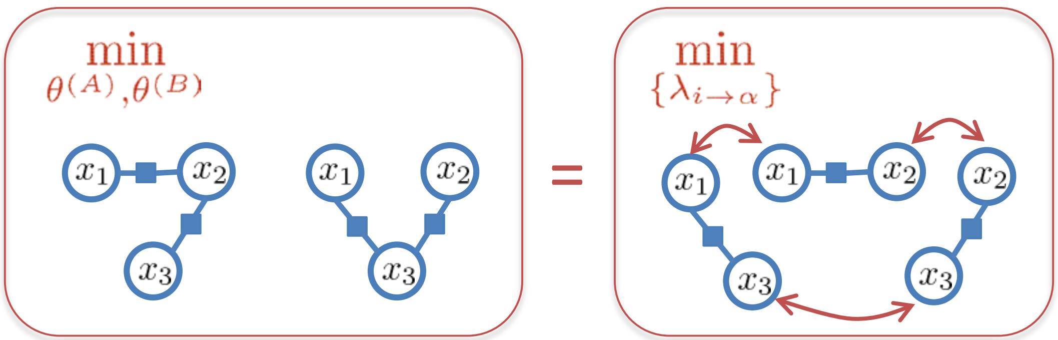
$$= \min_{\lambda} \max_{\mu} \sum_{i,k} \Big(\theta_{i;k} - \sum_j \lambda_{i\to ij;k}\Big)\mu_{i;k} + \sum_{i,j,k,l} \big(\theta_{ij;kl} + \lambda_{i\to ij;k} + \lambda_{j\to ij;l}\big)\mu_{ij;kl}$$

$$= \min_{\lambda} \sum_{i,k} \max_k \Big(\theta_{i;k} - \sum_j \lambda_{i\to ij;k}\Big) + \sum_{i,j,k,l} \max_{k,l} \big(\theta_{ij;kl} + \lambda_{i\to ij;k} + \lambda_{j\to ij;l}\big)$$

Duality: MAP

Primal

$$\min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i)\right]$$

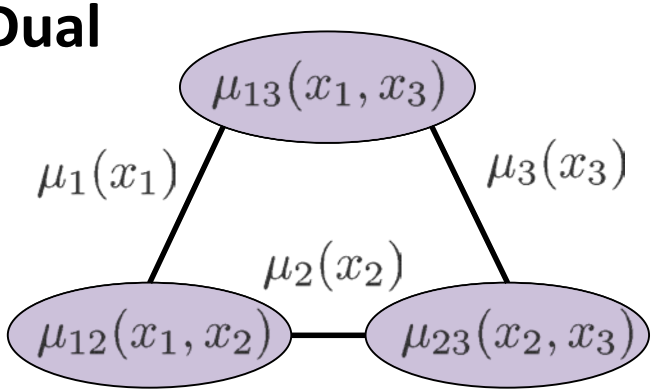


Reason about subproblems

“Messages” adjust overlapping subproblems

Reparameterize subproblems to decrease upper bound

Dual



$$\max_{\vec{\mu}\in\mathbb{L}} \vec{\theta}\cdot\vec{\mu}$$

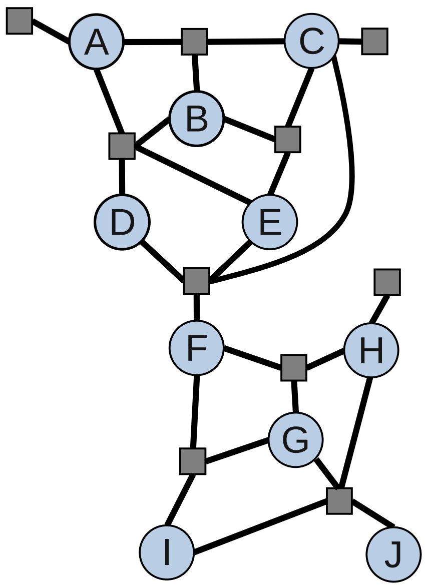
Reason about “beliefs” (marginals)

Constraints enforce overlapping beliefs are consistent

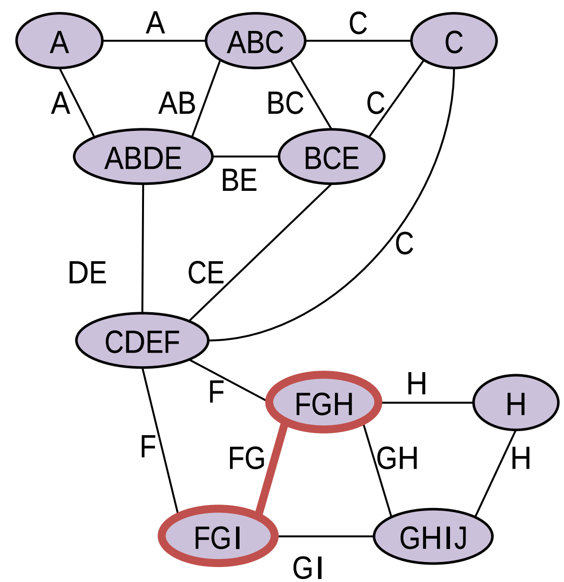
Optimum over beliefs gives upper bound

Regions

- Generalize local consistency enforcement



Factor graph



Dual graph

Beliefs: $\mu_{FGH}, \; \mu_{FGI}, \; \ldots$

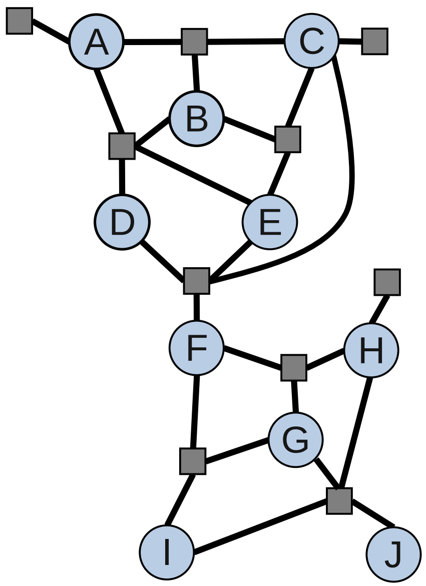
Consistency:

$$\sum_a \mu_{FGH}(f,g,h) = \mu_{FG}(f,g) = \sum_i \mu_{FGI}(f,g,i)$$

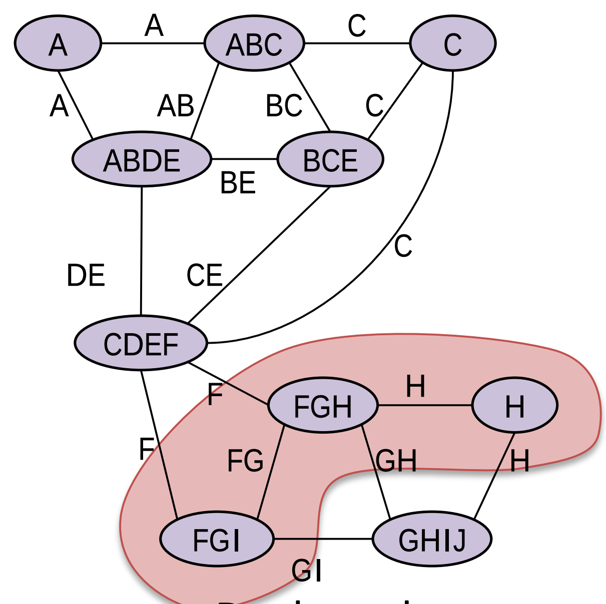
Separators = coordinates of bound optimization (¸)

Regions

- Generalize local consistency enforcement
- Larger regions: more consistent; more costly to represent



Factor graph



Dual graph

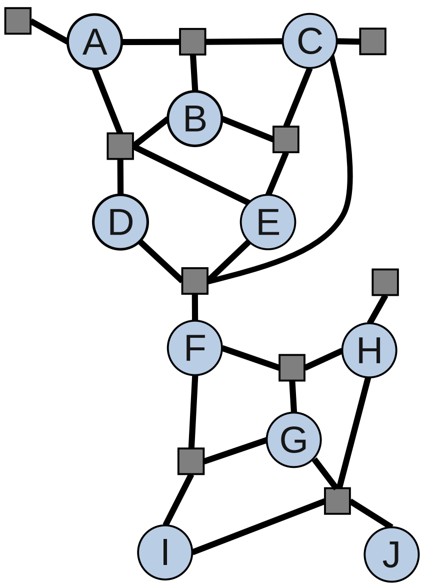
Beliefs: $\mu_{FGH}, \ \mu_{FGI}, \ \ldots$

Consistency:

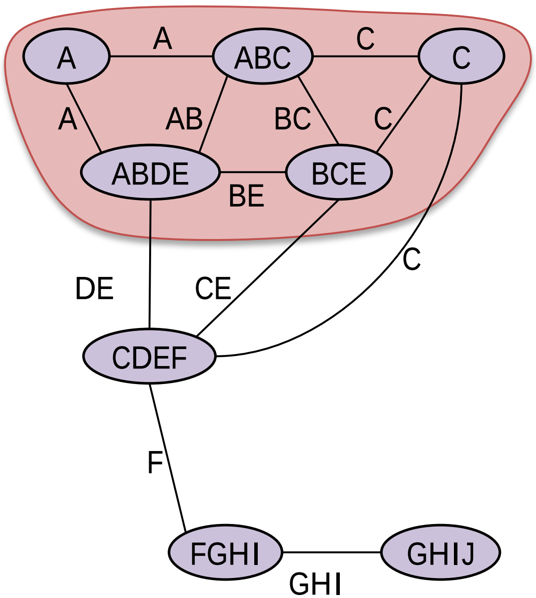
$$\sum_a \mu_{FGH}(f,g,h) = \mu_{FG}(f,g) = \sum_i \mu_{FGI}(f,g,i)$$

Regions

- Generalize local consistency enforcement
- Larger regions: more consistent; more costly to represent



Factor graph



Dual graph

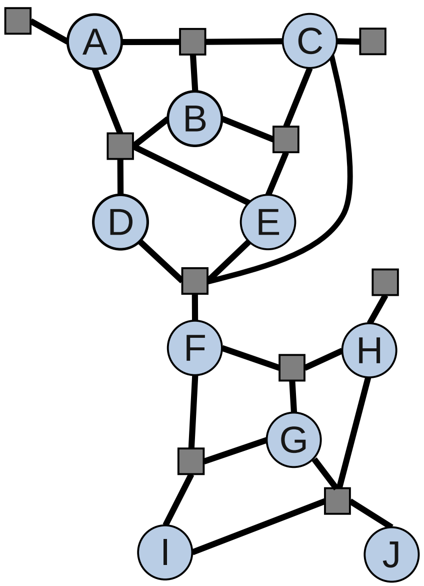
Beliefs: $\mu_{FGHI}, \; \ldots$

Consistency:

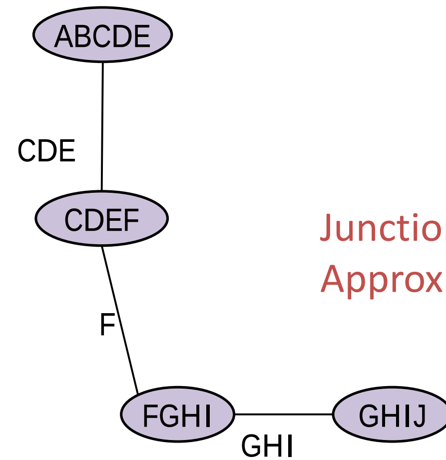
$$\sum_a \mu_{FGHI}(f,g,h,i) = \mu_{GHI}(g,h,i) = \ldots$$

Regions

- Generalize local consistency enforcement
- Larger regions: more consistent; more costly to represent



Factor graph



Dual graph

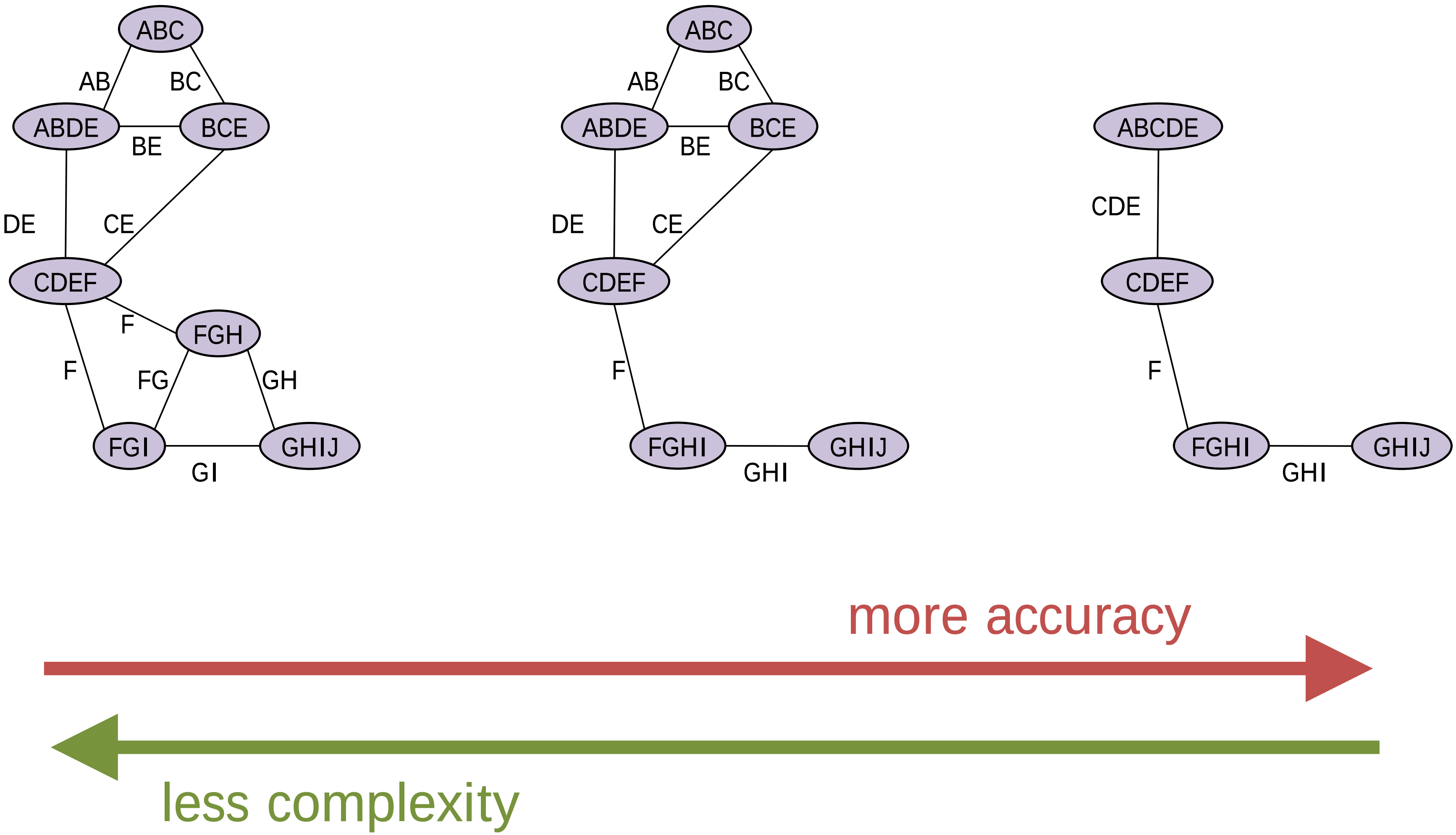
Junction tree:
Approximation is exact!

Beliefs: $\mu_{FGHI}, \ldots$

Consistency:

$$\sum_a \mu_{FGHI}(f,g,h,i) = \mu_{GHI}(g,h,i) = \ldots$$

Regions


ABC
AB
BC
ABDE
BCE
BE
DE
CE
CDEF
F
FGH
F
FG
GH
FGI
GHIJ
GI
ABC
AB
BC
ABDE
BCE
BE
DE
CE
CDEF
F
FGHI
GHIJ
GHI
ABCDE
CDE
CDEF
F
FGHI
GHIJ
GHI
more accuracy
less complexity

Mini-bucket Regions

- Mini-bucket elimination defines regions with bounded complexity

mini-buckets

B: $\underbrace{f(a,b)\ f(b,c)}$ $\underbrace{f(b,d)\ f(b,e)}$

C: $f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c)$

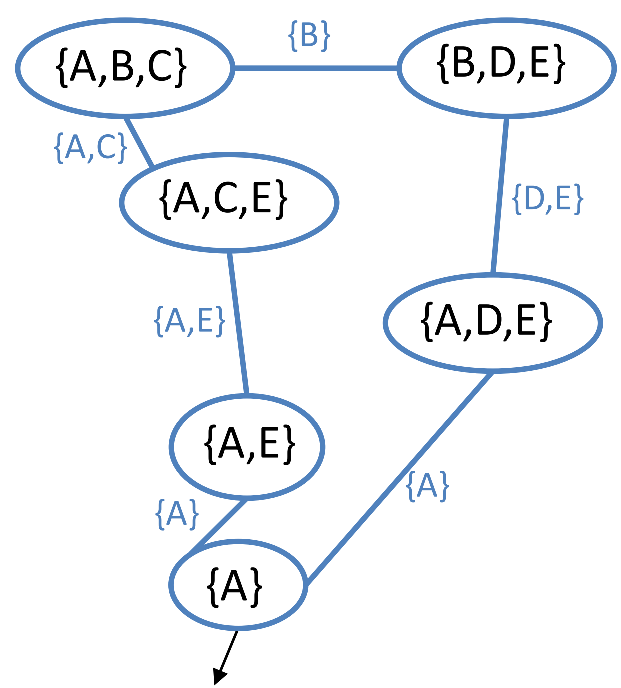
D: $f(a,d)\ \lambda_{B\to D}(d,e)$

E: $\lambda_{C\to E}(a,e)\ \lambda_{E\to E}(a,e)$

A: $f(a)\ \lambda_{E\to A}(a)$

U = upper bound

Join graph:



U = upper bound

Variational perspectives

- Replace q 2 P and H(q) with simpler approximations

$$\log p(x^*) = \max_{q \in \mathbb{P}} \mathbb{E}_q[\log f(x)]$$

$$\log Z = \max_{q \in \mathbb{P}} \mathbb{E}_q[\log f(x)] + \boxed{H(x\,;\,q)}$$

Approximate entropy in terms of local beliefs

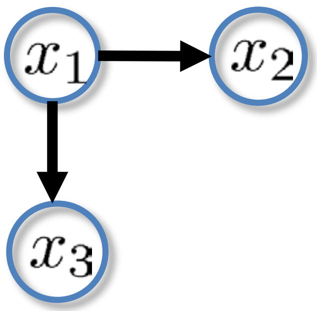
- Algorithms and their properties:

	Method	distributions	entropy	value
Max:	Linear programming	$q \in \mathbb{L} \supseteq \mathbb{P}$	n/a	$\hat{p}_{lp} \geq p(x^*)$
Sum:	Mean field	$\{q = \prod q_i(x_i)\} \subseteq \mathbb{P}$	exact	$Z_{mf} \leq Z$
	Belief propagation	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_\beta \approx H(q)$	$Z_\beta \approx Z$
	Tree-reweighted	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_{tr} \geq H(q)$	$Z_{tr} \geq Z$

Bethe Approximation

- Need to approximate H in terms of only local beliefs

- In trees, H has a simple form:



$$\begin{aligned} H(p) &= -\mathbb{E}\Big[\log\ p(x_1)p(x_2|x_1)p(x_3|x_1)\Big] \\ &= -\mathbb{E}\Big[\log\ p(x_1)\frac{p(x_2,x_1)}{p(x_1)}\,\frac{p(x_3,x_1)}{p(x_1)}\Big] \\ &= -\mathbb{E}\Big[\log\ p(x_1)\,p(x_2)\,p(x_3)\frac{p(x_2,x_1)}{p(x_1)p(x_2)}\,\frac{p(x_3,x_1)}{p(x_1)p(x_3)}\Big] \end{aligned}$$

Then, $H(p) = = \sum_i H[p(x_i)] - \sum_{ij \in E} \mathbb{I}[p(x_i, x_j)]$

Depends only on pairwise marginals!

Called the “Bethe” approximation in statistical physics

see [Yedidia et al. 2001]

Bethe Approximation

- Suppose we want to optimize

$$\max_{b \in \mathbb{L}} \sum_{\alpha} \mathbb{E}_{b_\alpha}[\theta_\alpha(x_\alpha)] + \sum_i \mathbb{H}(b_i) - \sum_{ij} \mathbb{I}(b_{ij})$$

$$\mathbb{L}_G = \{b_i, b_{ij} : \sum_{x_i} b_{ij}(x_i, x_j) = b_j(x_j), \sum_{x_j} b_j(x_j) = 1, b_{ij} \geq 0\}$$

- Use the same Lagrange multiplier trick as LP/DD
  - Then, define $m_{i \to \alpha}(x_i) \propto \exp[\lambda_{i \to \alpha}(x_i)]$

Fixed points satisfy LBP recursion!

Calculating messages:

$$m_{i \to \alpha}(x_i) \propto \prod_{\beta \neq \alpha} m_{\beta \to i}(x_i)$$

$$m_{\alpha \to i}(x_i) \propto \sum_{x_\alpha \setminus x_i} f_\alpha(x_\alpha) \prod_{j \neq i} m_{j \to \alpha}(x_j)$$

Calculating marginals:

$$b(x_i) \propto \prod_{\alpha \ni i} m_{\alpha \to i}(x_i)$$

$$b(x_\alpha) \propto f_\alpha(x_\alpha) \prod_{i \in \alpha} m_{i \to \alpha}(x_i)$$

Loopy BP and the partition function

- Use the Bethe approximation to estimate log Z:
  - Run loopy BP on the factor graph & calculate beliefs
  - Use the Bethe approximation to H(b):

$$\log Z \approx \sum_\alpha \mathbb{E}_{b_\alpha}[\log f_\alpha(x_\alpha)] + \sum_i \mathbb{H}(b_i) - \sum_\alpha \mathbb{E}_{b_\alpha}\Big[\log \frac{b_\alpha}{\prod_{i \in \alpha} b_i}\Big]$$

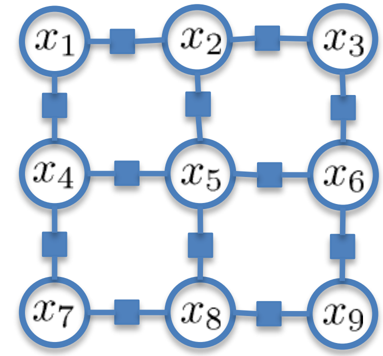
  - Often written using counting numbers:

$$\log Z \approx \sum_\alpha \mathbb{E}_{b_\alpha}[\log f_\alpha(x_\alpha)] + \sum_\alpha c_\alpha H(b_\alpha) + \sum_i c_i H(b_i)$$

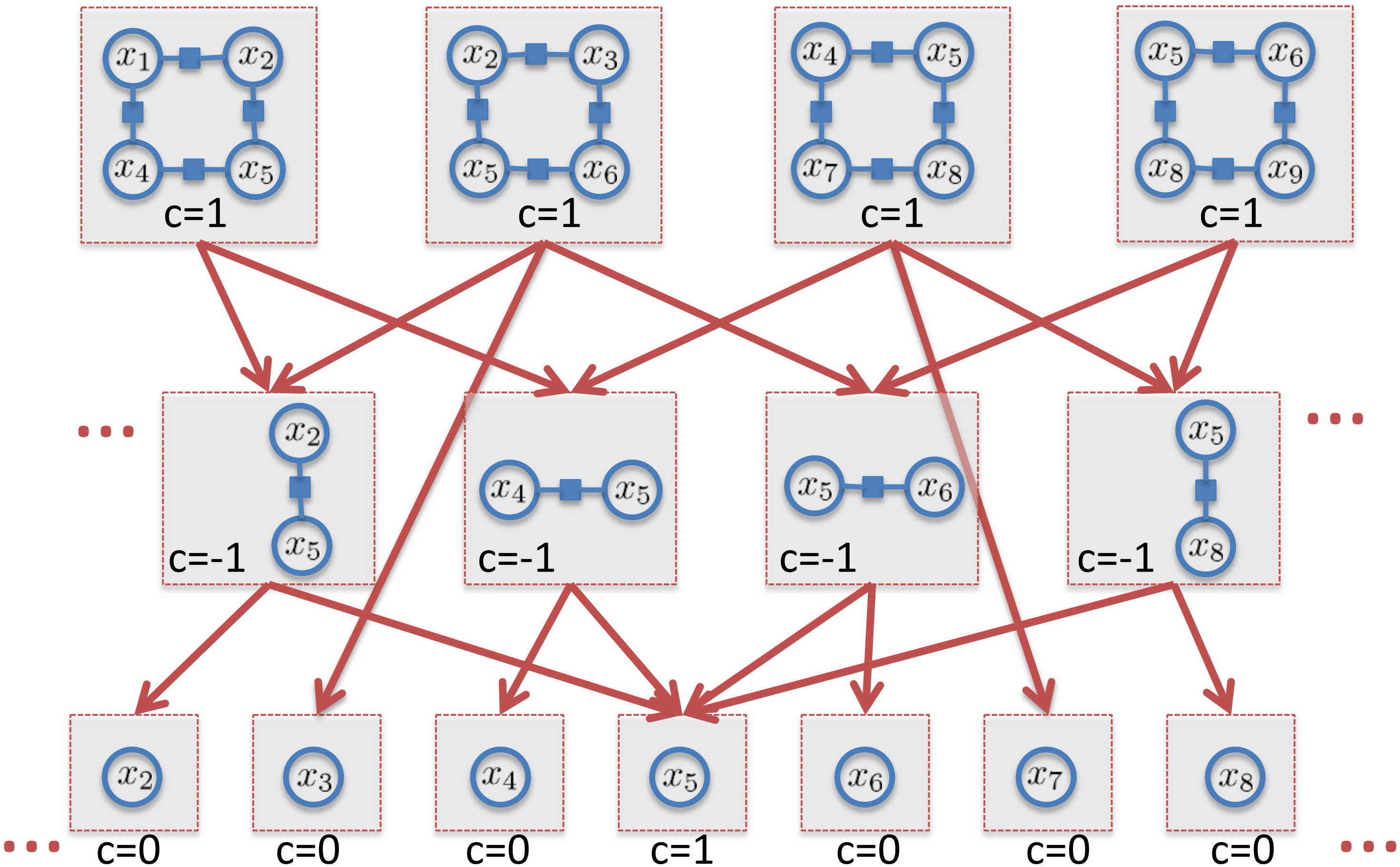
$$c_\alpha = 1, \quad c_i = 1 - \deg(i)$$

  - As with LP / DD, regions are what matters!
  - But now, regions define both **consistency** and **entropy**

Region graphs



Region: a collection of variables & their interactions



Counting numbers:

$$c_\alpha = 1 - \sum_{\beta \supset \alpha} c_\beta$$

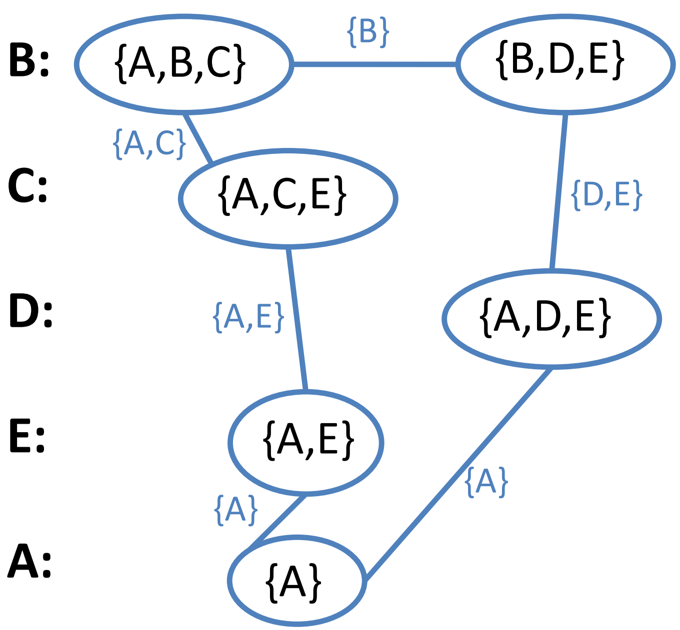
(inclusion/exclusion)

Join Graphs

- Join graphs give a simple set of regions

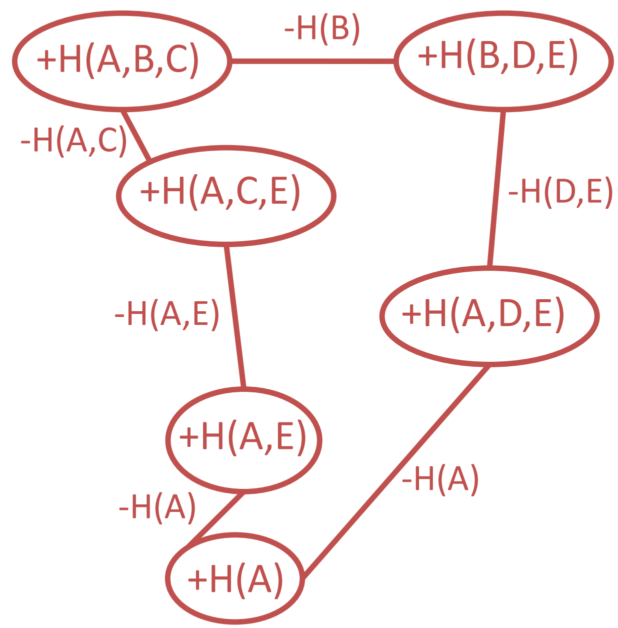
$$\log Z = \max_{\vec{\mu} \in \mathcal{M}} \vec{\theta} \cdot \vec{\mu} + H(\vec{\mu})$$

Join graph:



Each variable's subgraph is a tree

Entropy approximation:



Counting numbers
cliques: +1
separators: -1

Results in a simple variant of LBP message passing!

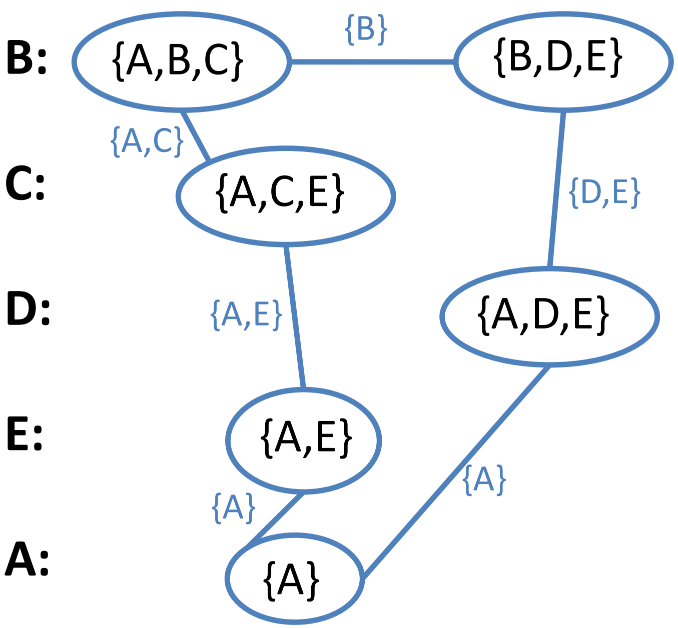
$$m_{\alpha \to \beta}(x_{\beta \setminus \alpha}) \propto \sum_{x_{\alpha \setminus \beta}} f_\alpha(x_\alpha) \prod_{\gamma \neq \beta} m_{\gamma \to \alpha}(x_\alpha)$$

Summation Bounds

- A local bound on the entropy will give a bound on Z:

$$\log Z = \max_{\vec{\mu} \in \mathcal{M}} \vec{\theta} \cdot \vec{\mu} + H(\vec{\mu})$$

Join graph:



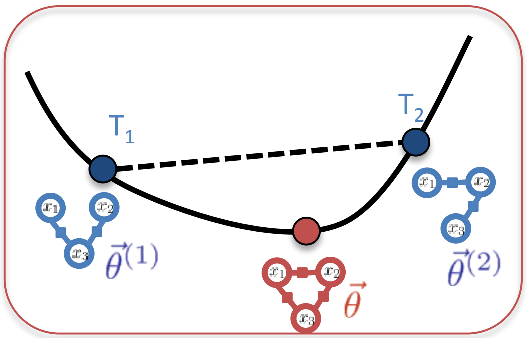
Exact Entropy

$$
\begin{array}{ccc}
H(B|A,C,D,E) & \cdot & w_1\, H(B|A,C) + w_2\, H(B|D,E) \\
+ & & + \\
H(C|A,D,E) & \cdot & H(C|A,E) \\
+ & & + \\
H(D|A,E) & = & H(D|A,E) \\
+ & & + \\
H(E|A) & = & H(E|A) \\
+ & & + \\
H(A) & = & H(A)
\end{array}
$$

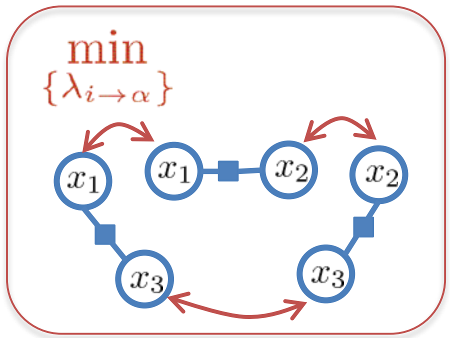
Weighted Mini-bucket (primal) [Liu & Ihler 2011]
Conditional Entropy Decomposition (dual) [Globerson & Jaakkola 2008]

Primal vs. Dual Forms

Primal $\Phi_\tau(\vec\theta) \le \min_{\{\lambda\}} \sum \Phi_{w_r}(\theta^{(r)} + \lambda^{(r)})$



or

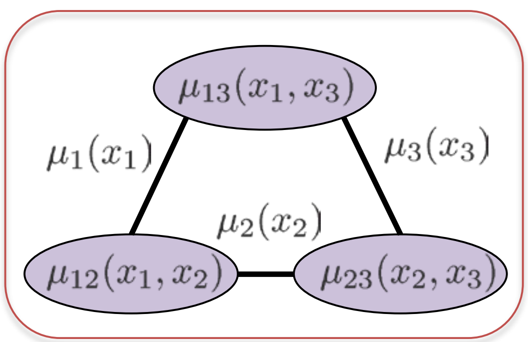


Direct bound on objective

“Messages” reparameterize
subproblems to be consistent

“Typically”:
upper bound: prefer primal
lower bound: either OK
Bethe / BP: prefer dual

Dual $\Phi_\tau(\vec\theta) \le \max_{\{\mu\}} \vec\theta \cdot \mu + \hat H(\mu)$



Reason about “beliefs” (marginals)

Messages update beliefs to be consistent

Message-passing form:

$$m_{ij}(x_j) \propto \left[\sum_{x_i} f_i(x_i)\; f_{ij}(x_i, x_j)^{1/\rho_{ij}} \frac{\prod_k m_{ki}(x_i)}{m_{ji}(x_i)^{1/\rho_{ij}}} \right]^{\rho_{ij}}$$

Summary: Variational methods

- Build approximations via an optimization perspective
  - **Primal** form: decomposition into simpler problems
  - **Dual** form: optimization over local "beliefs"

- Deterministic bounds and approximations
  - Convex upper bounds
  - Non-convex lower bounds
  - Bethe approximation & belief propagation

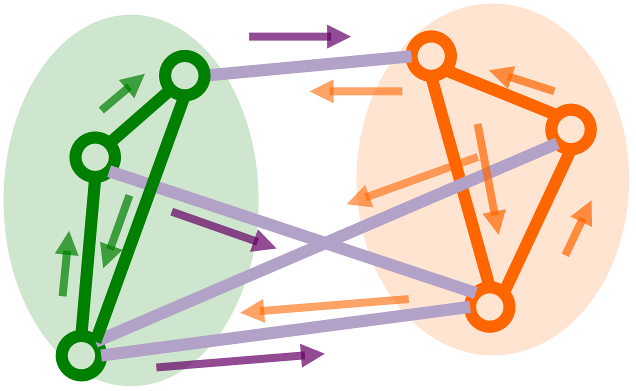
- Scalable, "local approximation" viewpoint
  - Optimization as local message passing

- Can improve quality through increasing region size
  - But, requires exponentially increasing memory & time

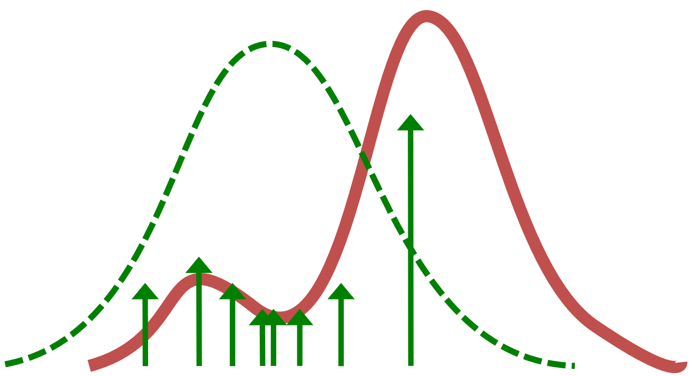
Outline

- Review: Graphical Models

- Variational methods
  - Convexity & decomposition bounds
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships



- **Monte Carlo sampling**
  - **Basics**
  - Importance sampling
  - Markov chain Monte Carlo
  - Integrating inference and sampling



Monte Carlo estimators

- Most basic form: empirical estimate of probability

$$\mathbb{E}[u(x)] = \int p(x)u(x) \quad \approx \quad U = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Relevant considerations
  - Able to sample from the target distribution p(x)?
  - Able to evaluate p(x) explicitly, or only up to a constant?

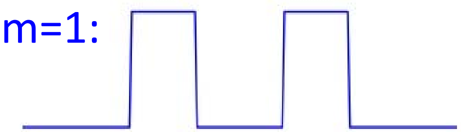
- "Any-time" properties
  - Unbiased estimator, $\mathbb{E}[U] = \mathbb{E}[u(x)]$
    or asymptotically unbiased, $\mathbb{E}[U] \to \mathbb{E}[u(x)]$ as $m \to \infty$
  - Variance of the estimator decreases with m

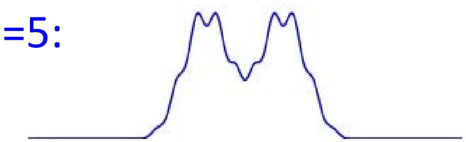
Monte Carlo estimators

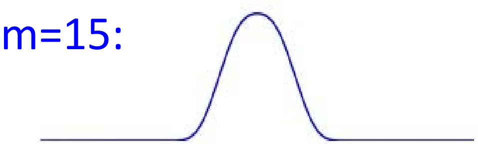
- Most basic form: empirical estimate of probability

$$\mathbb{E}[u(x)] = \int p(x) u(x) \quad \approx \quad U = \frac{1}{m} \sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Central limit theorem
  - p(U) is asymptotically Gaussian:

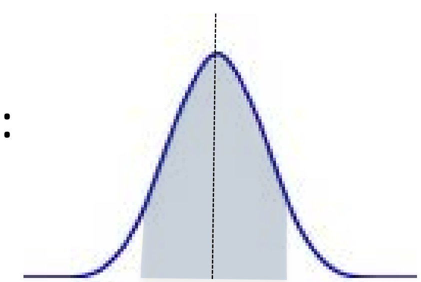
m=1: 

m=5: 

m=15: 

- Finite sample confidence intervals
  - If u(x) or its variance are bounded, e.g., $u(x^{(i)}) \in [0,1]$
    probability concentrates rapidly around the expectation:

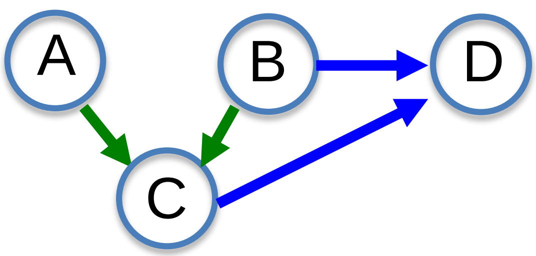
$$\Pr\big[\, |U - \mathbb{E}[U]| > \epsilon \,\big] \leq O\big(\, \exp(-m\epsilon^2)\,\big)$$



Sampling in Bayes nets [e.g., Henrion 1988]

- No evidence: "causal" form makes sampling easy
  - Follow variable ordering defined by parents
  - Starting from root(s), sample downward
  - When sampling each variable, condition on values of parents

$$p(A,B,C,D) = p(A)\ p(B)\ p(C \mid A,B)\ p(D \mid B,C)$$



Sample:

$$a \sim p(A)$$
$$b \sim p(B)$$
$$c \sim p(C \mid A=a, B=b)$$
$$d \sim p(D \mid C=c, B=b)$$

Bayes nets with evidence

- Estimating the probability of evidence, P[E=e]:

$$\mathrm{P}[E=e] = \mathbb{E}[\,\mathbb{1}[E=e]\,] \approx \quad U = \frac{1}{m}\sum_i \mathbb{1}[\tilde{e}^{(i)} = e]$$

  - Finite sample bounds: u(x) 2 [0,1] [e.g., Hoeffding]

$$\Pr\Big[|U - \mathbb{E}[U]| > \epsilon\Big] \le 2\exp(-2m\epsilon^2)$$

What if the evidence is unlikely? P[E=e]=1e-6) could estimate U = 0!

  - Relative error bounds [Dagum & Luby 1997]

$$\Pr\Big[\frac{|U - \mathbb{E}[U]|}{\mathbb{E}[U]} > \epsilon\Big] \le \delta \quad \text{if} \quad m \ge \frac{4}{\mathbb{E}[U]\epsilon^2}\log\frac{2}{\delta}$$

Bayes nets with evidence

- Estimating posterior probabilities, P[A = a | E=e]?

- Rejection sampling
  - Draw x ~ p(x), but discard if E != e
  - Resulting samples are from p(x | E=e); use as before
  - Problem: keeps only P[E=e] fraction of the samples!
  - Performs poorly when evidence probability is small

- Estimate the ratio: P[A=a,E=e] / P[E=e]
  - Two estimates (numerator & denominator)
  - Good finite sample bounds require low *relative* error!
  - Again, performs poorly when evidence probability is small

Exact sampling via inference

- Draw samples from P[A|E=e] directly?
  - Model defines un-normalized p(A,…,E=e)
  - Build (oriented) tree decomposition & sample

$$\tilde{\mathbf{b}} \sim f(\tilde a, b)\cdot f(b,\tilde c)\cdot f(b,\tilde d)\cdot f(b,\tilde e)/\lambda_{B\to C}$$

$$\tilde{\mathbf{c}} \sim f(c,\tilde a)\cdot f(c,\tilde e)\cdot \lambda_{B\to C}(\tilde a, c, \tilde d, \tilde e)/\lambda_{C\to D}$$

$$\tilde{\mathbf{d}} \sim f(\tilde a, d)\cdot \lambda_{B\to D}(d,\tilde e)/\lambda_{D\to E}(\tilde a,\tilde e)$$

$$\tilde{\mathbf{e}} \sim \lambda_{D\to E}(\tilde a, e)/\lambda_{E\to A}(\tilde a)$$

$$\tilde{\mathbf{a}} \sim p(A) = f(a)\cdot\lambda_{E\to A}(a)/Z$$

B: $\overbrace{f(a,b)\ f(b,c)\ f(b,d)\ f(b,e)}^{\sum_b \prod}$

C: $f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c,d,e)$

D: $f(a,d)\ \lambda_{C\to D}(d,e,a)$

E: $\lambda_{D\to E}(a,e)$

A: $f(a)\ \lambda_{E\to A}(a)$

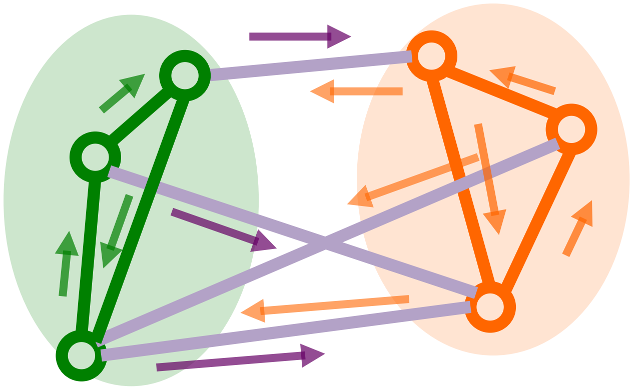
Z

Downward message normalizes bucket;
ratio is a conditional distribution

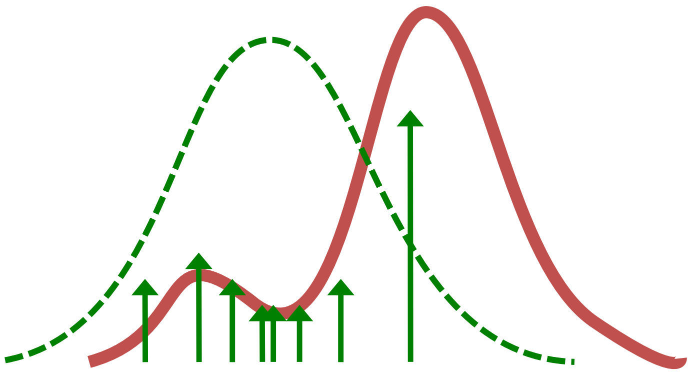
Work: O(exp(w)) to build distribution
O(n d) to draw each sample

Outline

- Review: Graphical Models
- Variational methods
  - Convexity & decomposition bounds
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships



- **Monte Carlo sampling**
  - Basics
  - **Importance sampling**
  - Markov chain Monte Carlo
  - Integrating inference and sampling



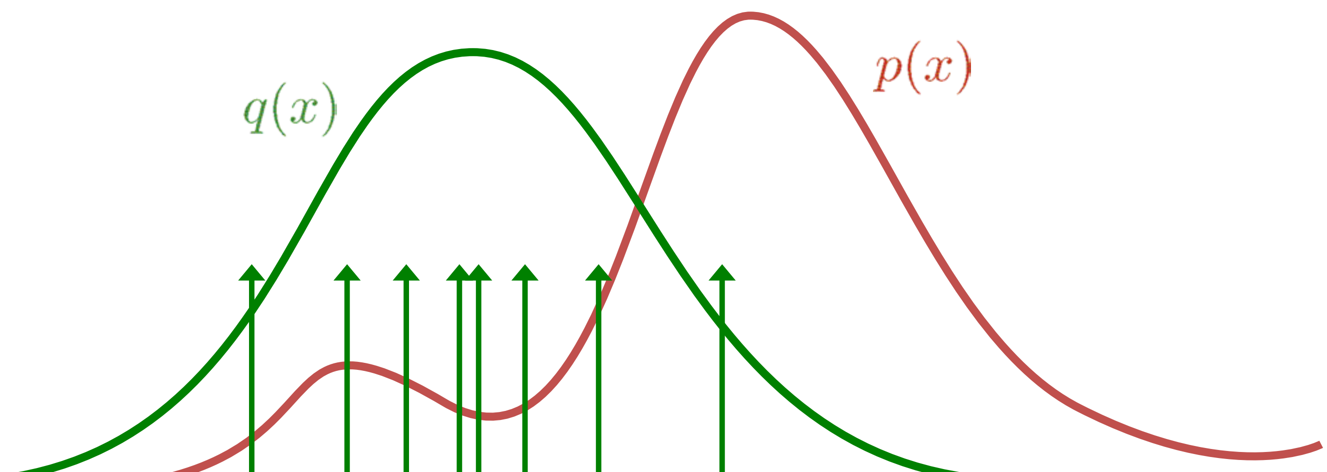
Importance Sampling

- Basic empirical estimate of probability:

$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$



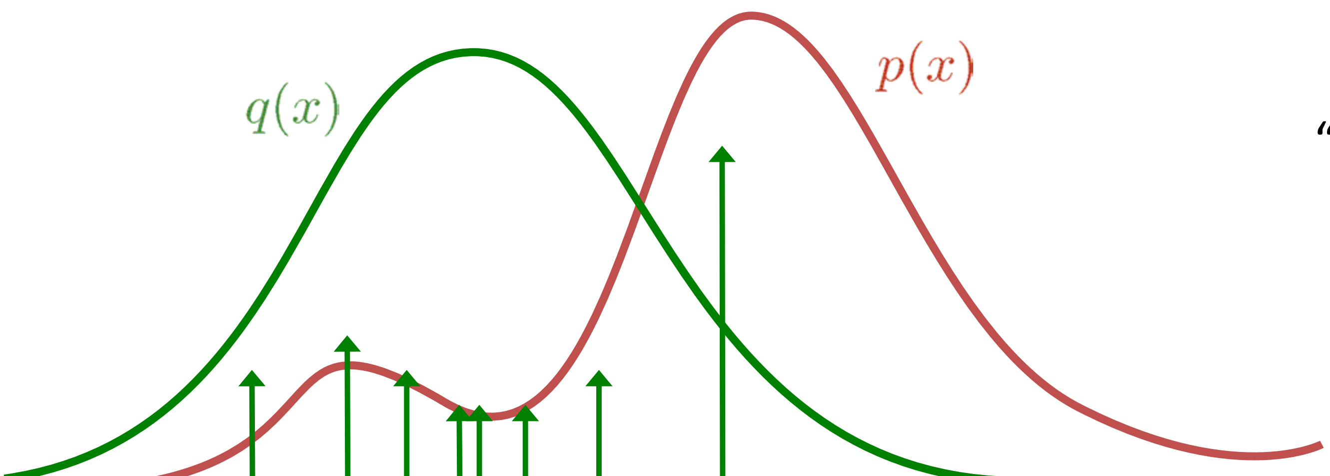
Importance Sampling

- Basic empirical estimate of probability:

$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$



"importance weights"

$$w^{(i)} = \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}$$

IS for common queries

- Partition function
  - Ex: MRF, or BN with evidence

$$Z = \sum_x f(x) = \sum_x q(x) \frac{f(x)}{q(x)} = \mathbb{E}_q\left[\frac{f(x)}{q(x)}\right] \approx \frac{1}{m} \sum w^{(i)}$$

  - Unbiased; only requires evaluating unnormalized function f(x)

$$w^{(i)} = \frac{f(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}$$

- General expectations wrt p(x) / f(x)?
  - E.g., marginal probabilities, etc.

$$\mathbb{E}_p[u(x)] = \sum_x u(x) \frac{f(x)}{Z} = \frac{\mathbb{E}_q[u(x) f(x)/q(x)]}{\mathbb{E}_q[f(x)/q(x)]} \approx \frac{\sum u(\tilde{x}^{(i)}) w^{(i)}}{\sum w^{(i)}}$$

Estimate separately

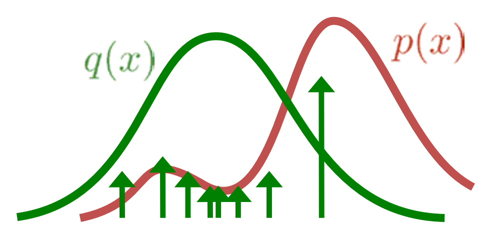
Only asymptotically unbiased…

Importance Sampling

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$

- IS is unbiased and fast if q(.) is easy to sample from
- IS can be lower variance if q(.) is chosen well
  - Ex: q(x) puts more probability mass where u(x) is large
  - Optimal: q(x) / |u(x) p(x)|
- IS can also give poor performance
  - If q(x) << u(x) p(x): rare but very high weights!
  - Then, empirical variance is also unreliable!
  - For guarantees, need to analytically bound weights / variance…



Choosing a proposal

[Liu, Fisher, Ihler 2015]

- Can use WMB upper bound to define a proposal q(x):

$$\tilde{\mathbf{b}} \sim w_1\, q_1(b|\tilde{a}, \tilde{c}) \;+\; w_2\, q_2(b|\tilde{d}, \tilde{e})$$

Weighted mixture:

use minibucket 1 with probability w_1

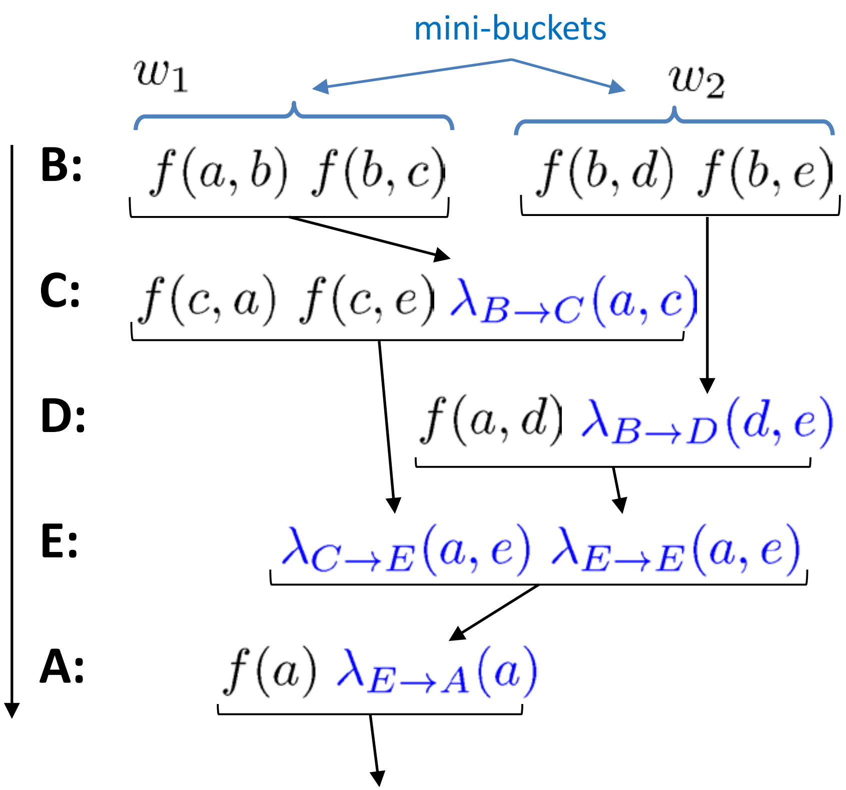
or, minibucket 2 with probability $w_2 = 1 - w_1$

where

$$q_1(b|a,c) = \left[\frac{f(a,b) \cdot f(b,c)}{\lambda_{B\to C}(a,c)}\right]^{\frac{1}{w_1}}$$

$\vdots$

$$\tilde{\mathbf{a}} \sim q(A) = f(a) \cdot \lambda_{E\to A}(a)/U$$



Key insight: provides bounded importance weights!

U = upper bound

$$0 \le \frac{F(x)}{q(x)} \le U \qquad \forall x$$

WMB-IS Bounds

[Liu, Fisher, Ihler 2015]

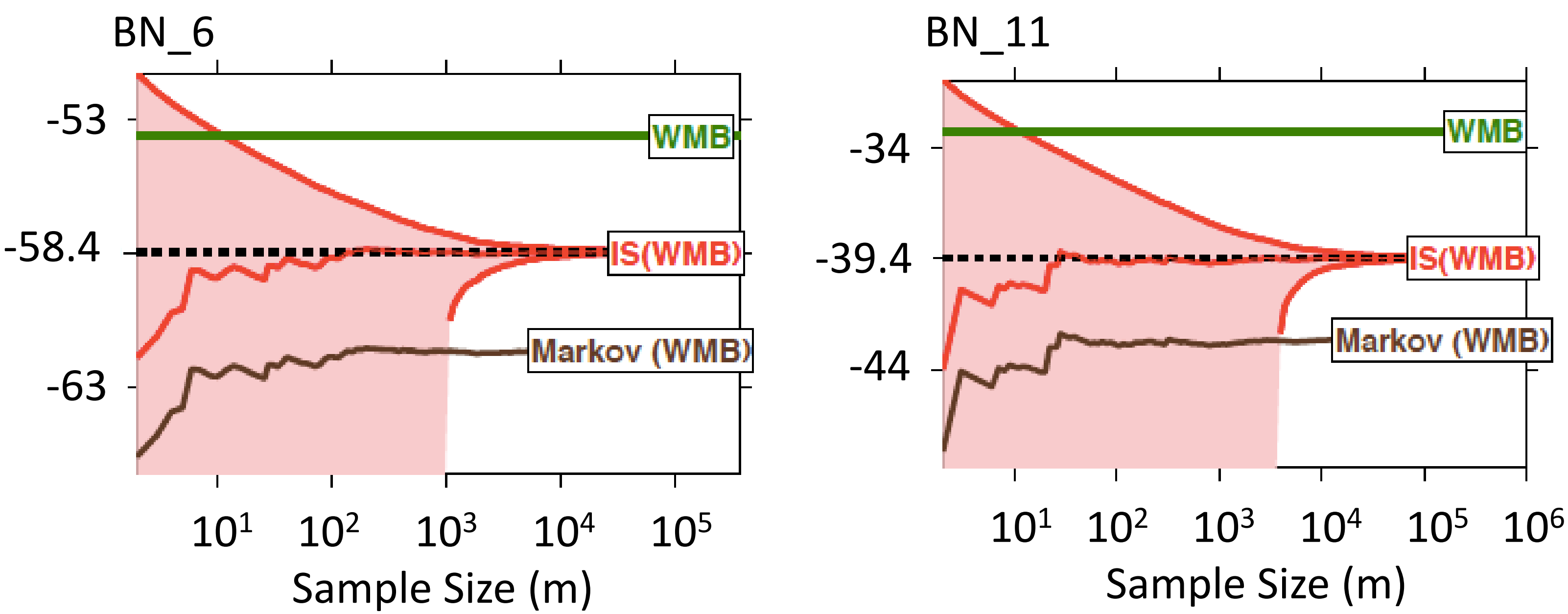
- Finite sample bounds on the average

$$\Pr\Big[|\hat{Z} - Z| > \epsilon\Big] \le 1 - \delta$$

$$\epsilon = \sqrt{\frac{2\hat{V}\log(4/\delta)}{m}} + \frac{7\,U\,\log(4/\delta)}{3(m-1)}$$

"Empirical Bernstein" bounds

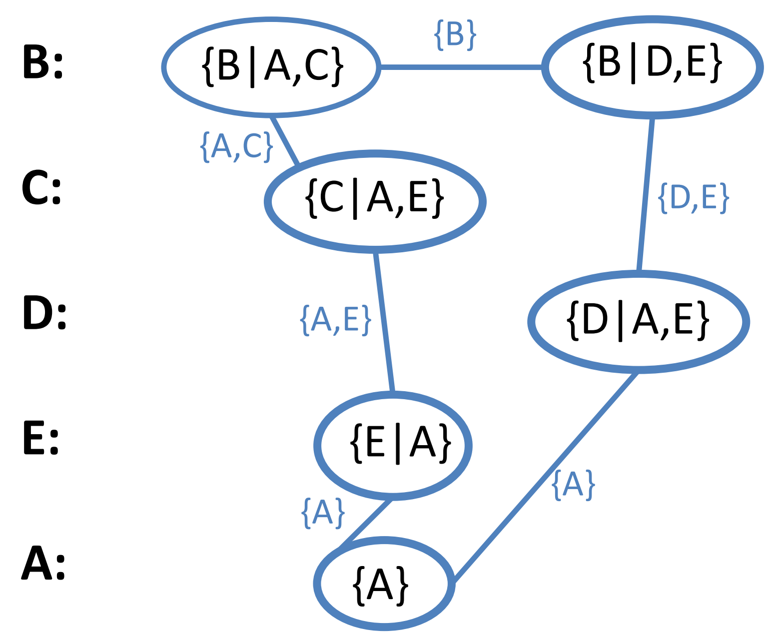
- Compare to forward sampling
  – Works well if evidence "not too unlikely") not too much less likely than U



Other choices of proposals

- Belief propagation
  - BP-based proposal [Changhe & Druzdzel 2003]
  - Join-graph BP proposal [Gogate & Dechter 2005]
  - Mean field proposal [Wexler & Geiger 2007]

Join graph:



Other choices of proposals

- Belief propagation
  - BP-based proposal [Changhe & Druzdzel 2003]
  - Join-graph BP proposal [Gogate & Dechter 2005]
  - Mean field proposal [Wexler & Geiger 2007]

- Adaptive importance sampling
  - Use already-drawn samples to update q(x)
  - Rates v_t and ´$_t$ adapt estimates, proposal
  - Ex:

    [Cheng & Druzdzel 2000]

    [Lapeyre & Boyd 2010]

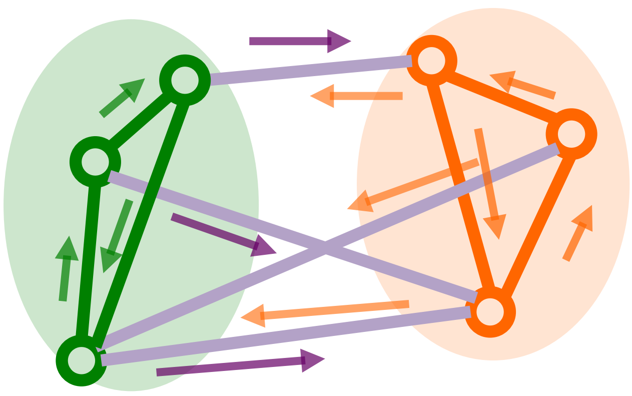
    …
  - Lose "iid"-ness of samples

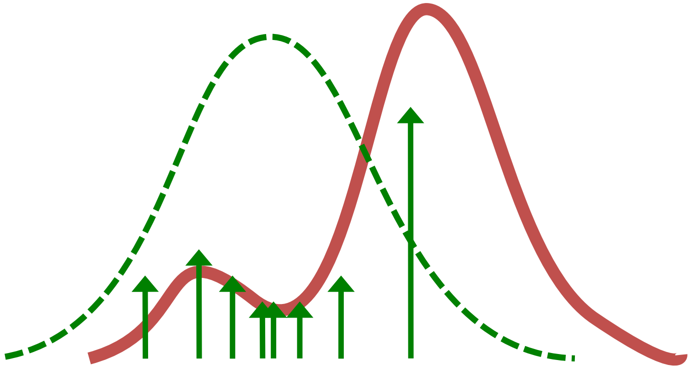
Adaptive IS

Initialize $q_0(x)$
for $t = 0 \ldots T$ **do**
  Draw $\tilde{X}_t = \{\tilde{x}^{(i)}\} \sim q_t(x)$
  $U_t = \frac{1}{m_t} \sum f(\tilde{x}^{(i)}) / q_t(\tilde{x}^{(i)})$
  $\hat{U} = (1 - v_t)\hat{U} + v_t U_t$
  $q_{t+1} = (1 - \eta_t) q_t + \eta_t q^*(X_t)$

Outline

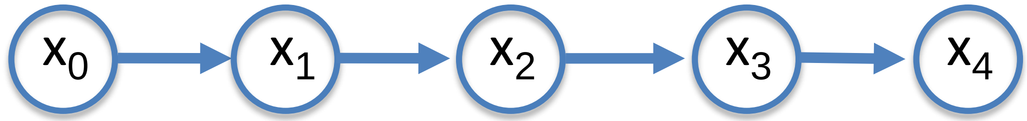
- Review: Graphical Models
- Variational methods
  - Convexity & decomposition bounds
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships
- **Monte Carlo sampling**
  - Basics
  - Importance sampling
  - **Markov chain Monte Carlo**
  - Integrating inference and sampling



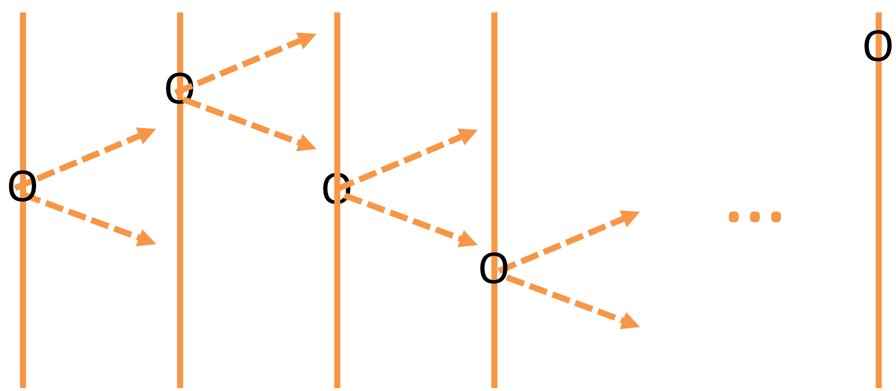


Markov Chains

- Temporal model
  - State at each time t
  - "Markov property": state at time t depends only on state at t-1
  - "Homogeneous" (in time): $p(X_t \mid X_{t-1}) = T(X_t \mid X_{t-1})$ does not depend on t

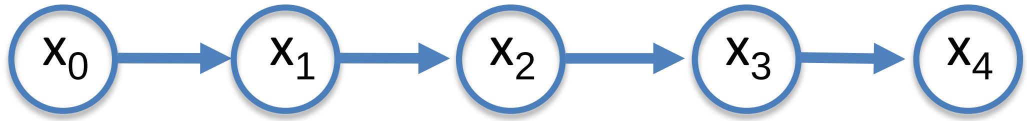


- Example: random walk
  - Time 0: $x_0 = 0$
  - Time t: $x_t = x_{t-1}$ § 1

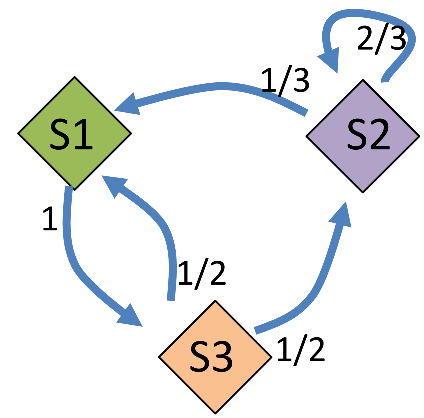


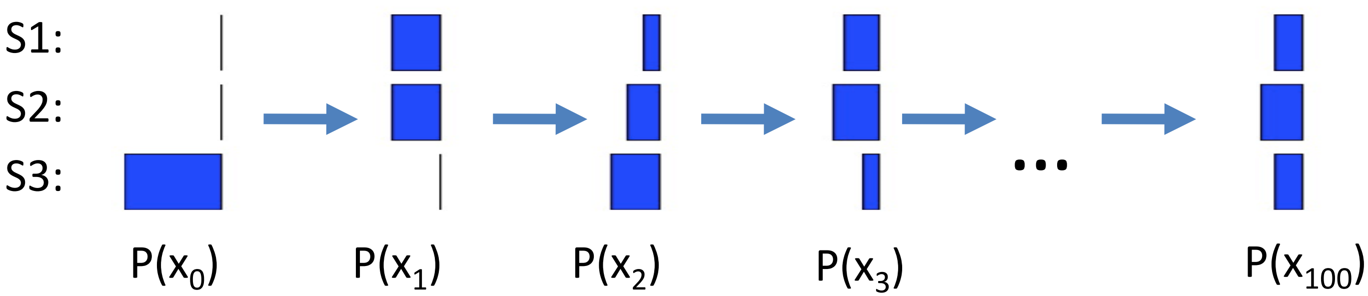
Markov Chains

- Temporal model
  - State at each time t
  - "Markov property": state at time t depends only on state at t-1
  - "Homogeneous" (in time): $p(X_t \mid X_{t-1}) = T(X_t \mid X_{t-1})$ does not depend on t



- Example: finite state machine
  - Time 0: x_0 = S3
  - Ex: S3 ! S1 ! S3 ! S2 ! ...
  - What is $p(x_t)$? Does it depend on x_0?



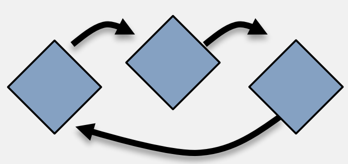


Stationary distributions

- Stationary distribution $s(x) \quad : \quad s(x_{t+1}) = \sum_{x_t} p(x_{t+1} \,|\, x_t) s(x_t)$

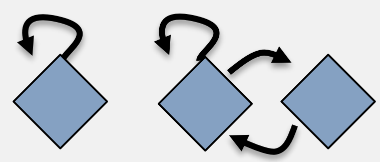
- $p(x_t)$ becomes independent of $p(x_0)$?
- Sufficient conditions for s(x) to exist and be unique:
  (a) p(. | .) is acyclic: $\gcd\{t \;:\; \Pr[x_t = s_i \,|\, x_0 = s_i] > 0\} = 1$
  (b) p(. | .) is irreducible: $\forall i, j \; \exists t \;:\; \Pr[x_t = s_i \,|\, x_0 = s_j] > 0$

Ex: not (a)



s(x) may not exist

Ex: not (b)

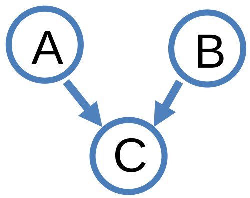


multiple s(x) exist

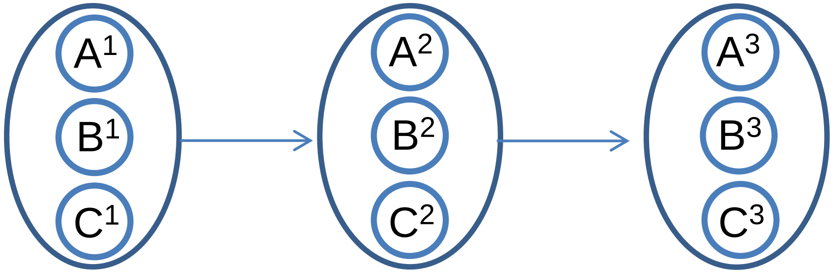
Without both (a) & (b), long-term probabilities may depend on the initial distribution

Markov Chain Monte Carlo

- Method for generating samples from an intractable p(x)
  - Create a Markov chain whose stationary distribution equals p(x)



State "x":
Complete config.
of target model



  - Sample $x^{(1)} \ldots x^{(m)}$; $x^{(m)} \sim p(x)$ if m sufficiently large
  - Two common methods:

- Metropolis sampling
  - Propose a new point x' using q(x' | x) ; depends on current point x
  - Accept with carefully chosen probability, a(x',x)
- Gibbs sampling
  - Sample each variable in turn, given values of all the others

Metropolis-Hastings

- At each step, propose a new value $x' \sim q(x'|x)$
- Decide whether we should move there
  - If $p(x') > p(x)$, it's a higher probability region (good)
  - If $q(x|x') < q(x'|x)$, it will be hard to move back (bad)

  - Accept move with a carefully chosen probability:

$$a(x', x) = \min\left[1\ ,\ \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right]$$

Probability of "accepting" the move from x to x'; otherwise, stay at state x.

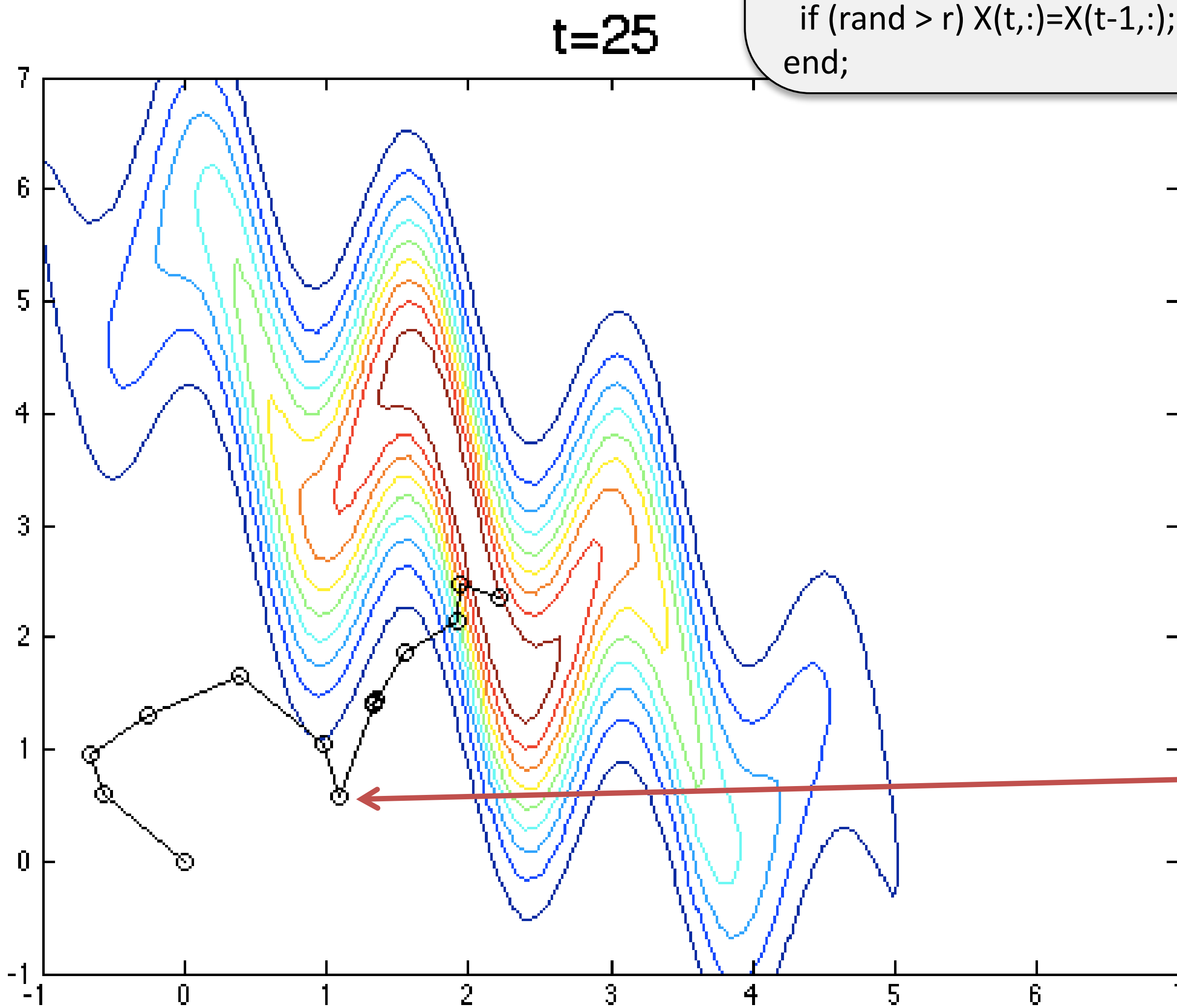
Ratio p(x') / p(x) means that we can substitute an unnormalized distribution f(x) if needed

  - The resulting transition probability $T(x'|x) = q(x'|x)\, a(x', x)$ has *detailed balance* with p(x), a sufficient condition for stationarity

MCMC Example

Metropolis-Hastings (symmetric proposal)

```
f = @(X) ...                              % define f(x) / p(x), target
X = [0,0];                                % set or sample initial state
for t=2:T,                                % simulate Markov chain:
  X(t,:) = X(t-1,:) + .5*randn(1,2);      % propose move
  r = min( 1, f(X(t,:)) / f(X(t-1,:)) );  % compute acceptance
  if (rand > r) X(t,:)=X(t-1,:); end;     % sample acceptance
end;
```



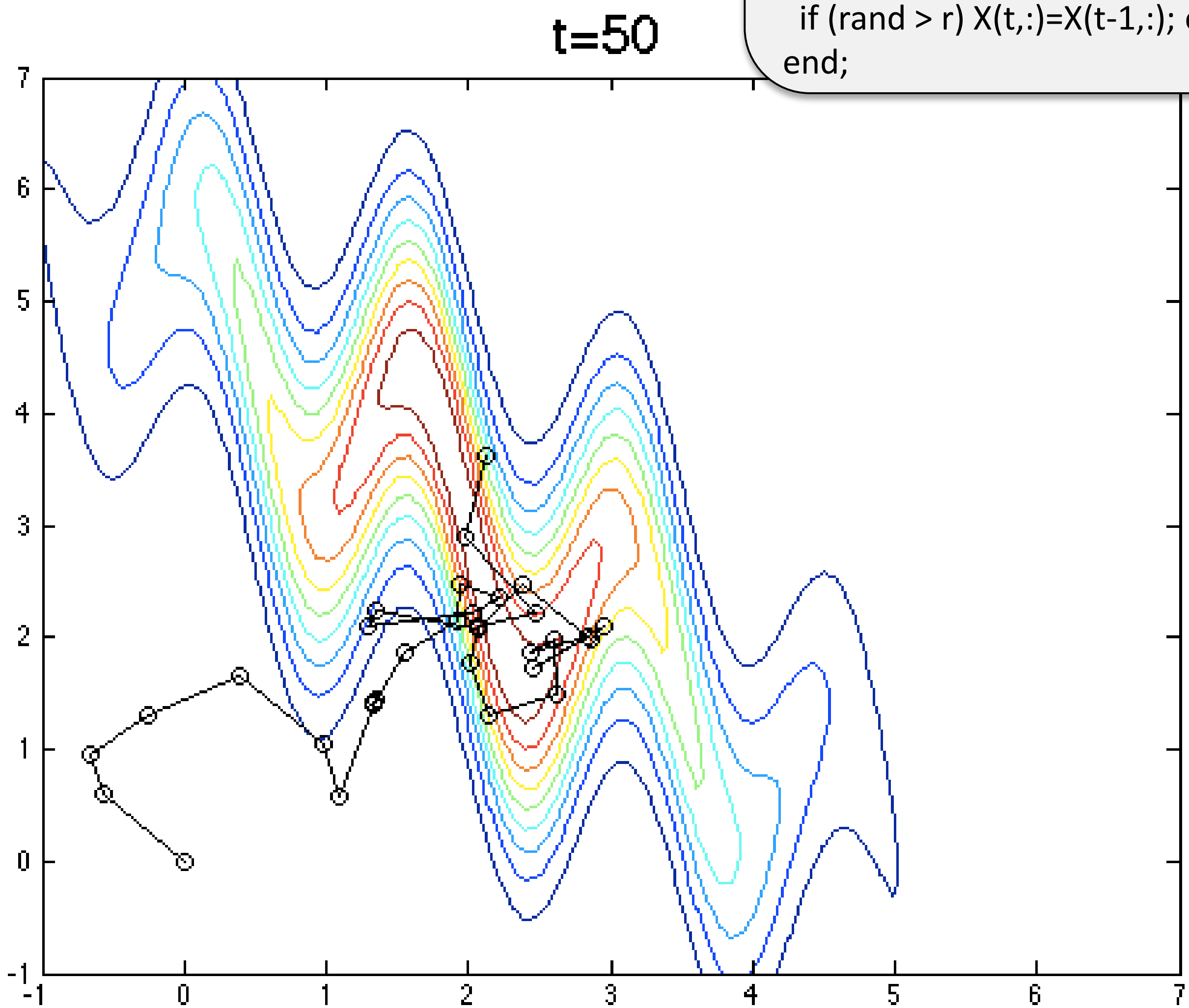
Early samples depend on initialization

"Burn in"; may discard these samples

MCMC Example

Metropolis-Hastings (symmetric proposal)

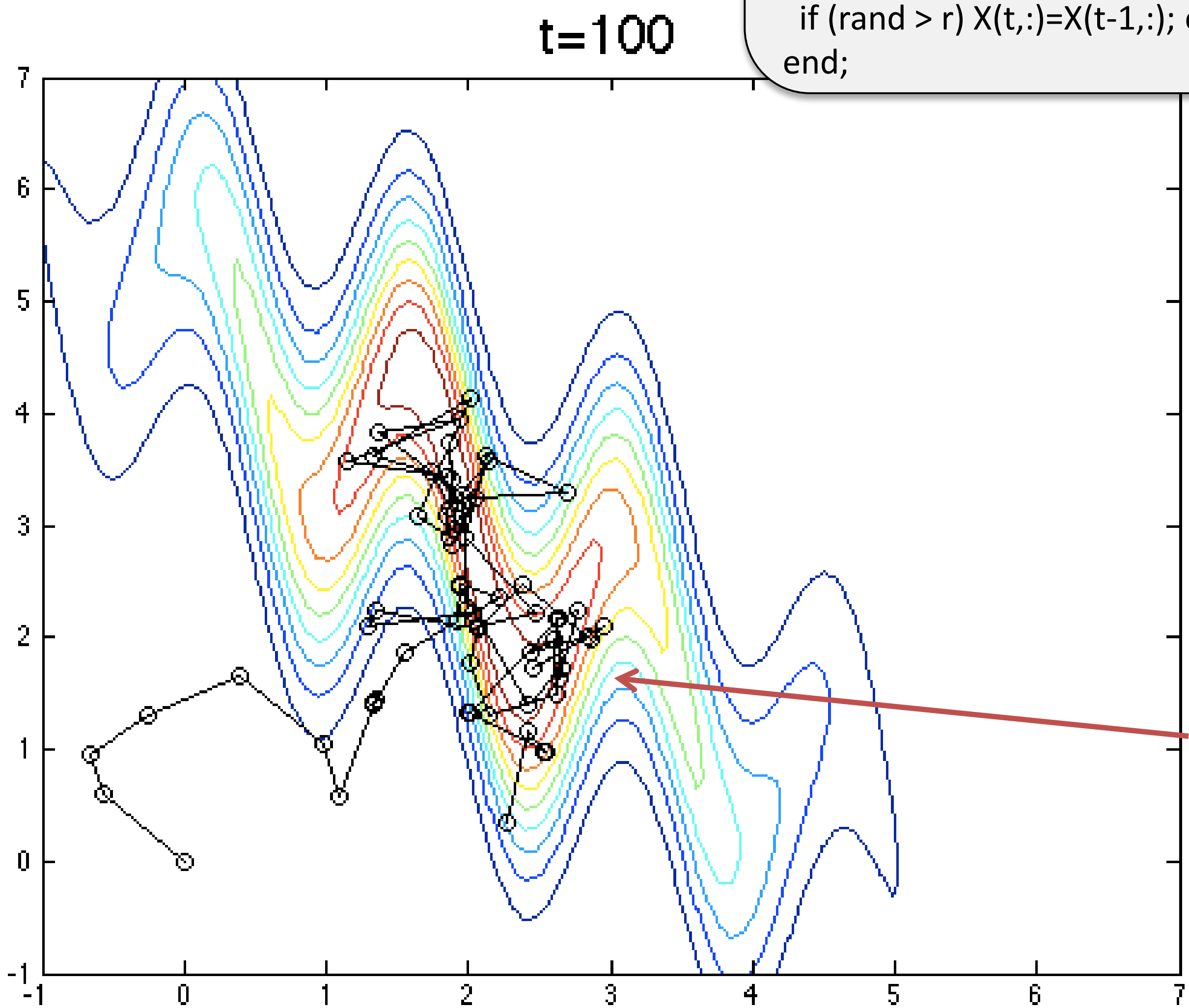
```
f = @(X) ...                    % define f(x) / p(x), target
X = [0,0];                      % set or sample initial state
for t=2:T,                      % simulate Markov chain:
  X(t,:) = X(t-1,:) + .5*randn(1,2);    % propose move
  r = min( 1, f(X(t,:)) / f(X(t-1,:)) );  % compute acceptance
  if (rand > r) X(t,:)=X(t-1,:); end;    % sample acceptance
end;
```



MCMC Example

Metropolis-Hastings (symmetric proposal)

```
f = @(X) ...                              % define f(x) / p(x), target
X = [0,0];                                % set or sample initial state
for t=2:T,                                % simulate Markov chain:
  X(t,:) = X(t-1,:) + .5*randn(1,2);     % propose move
  r = min( 1, f(X(t,:)) / f(X(t-1,:)) );  % compute acceptance
  if (rand > r) X(t,:)=X(t-1,:); end;     % sample acceptance
end;
```

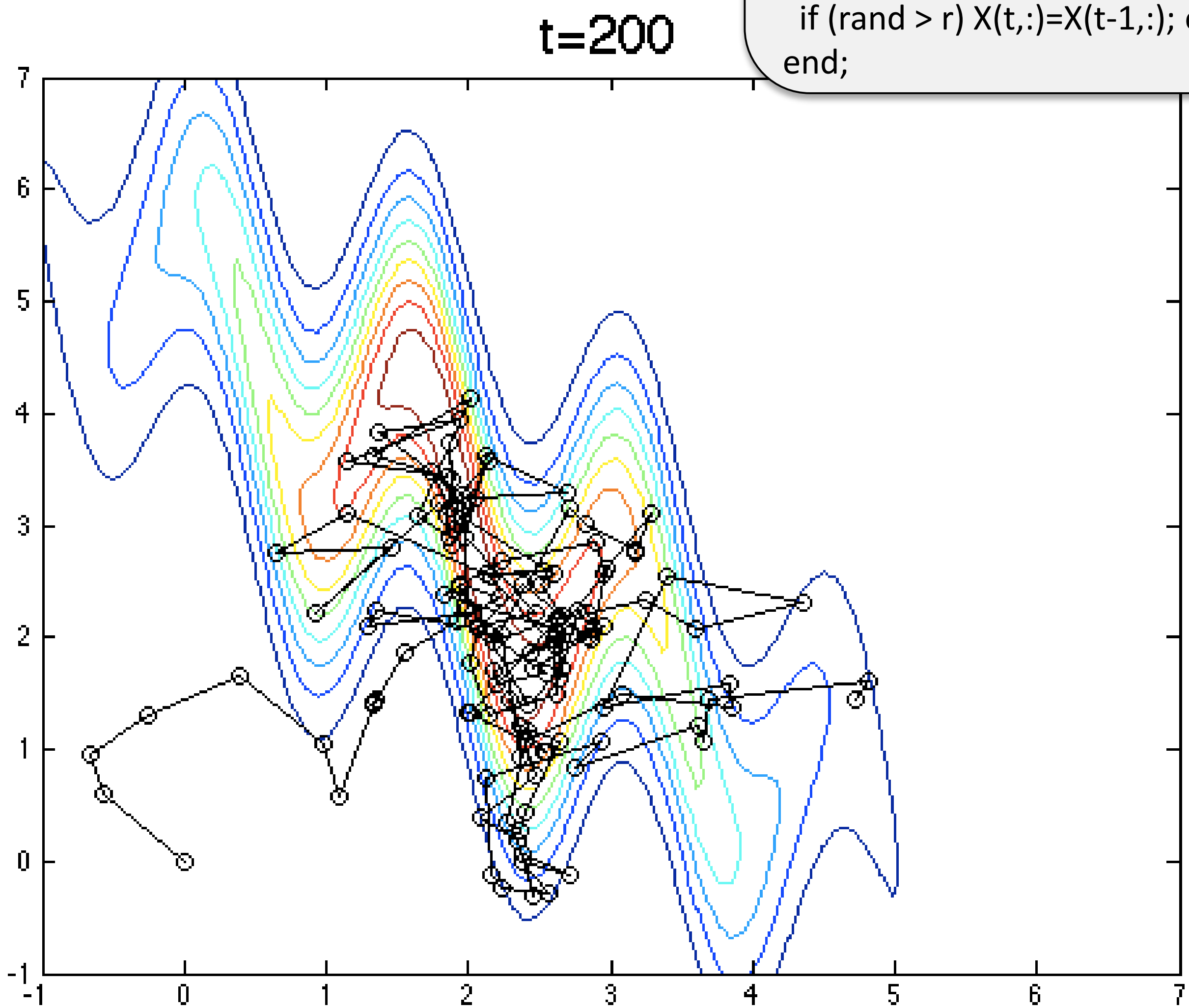


Samples correlated in time (not independent)

MCMC Example

Metropolis-Hastings (symmetric proposal)

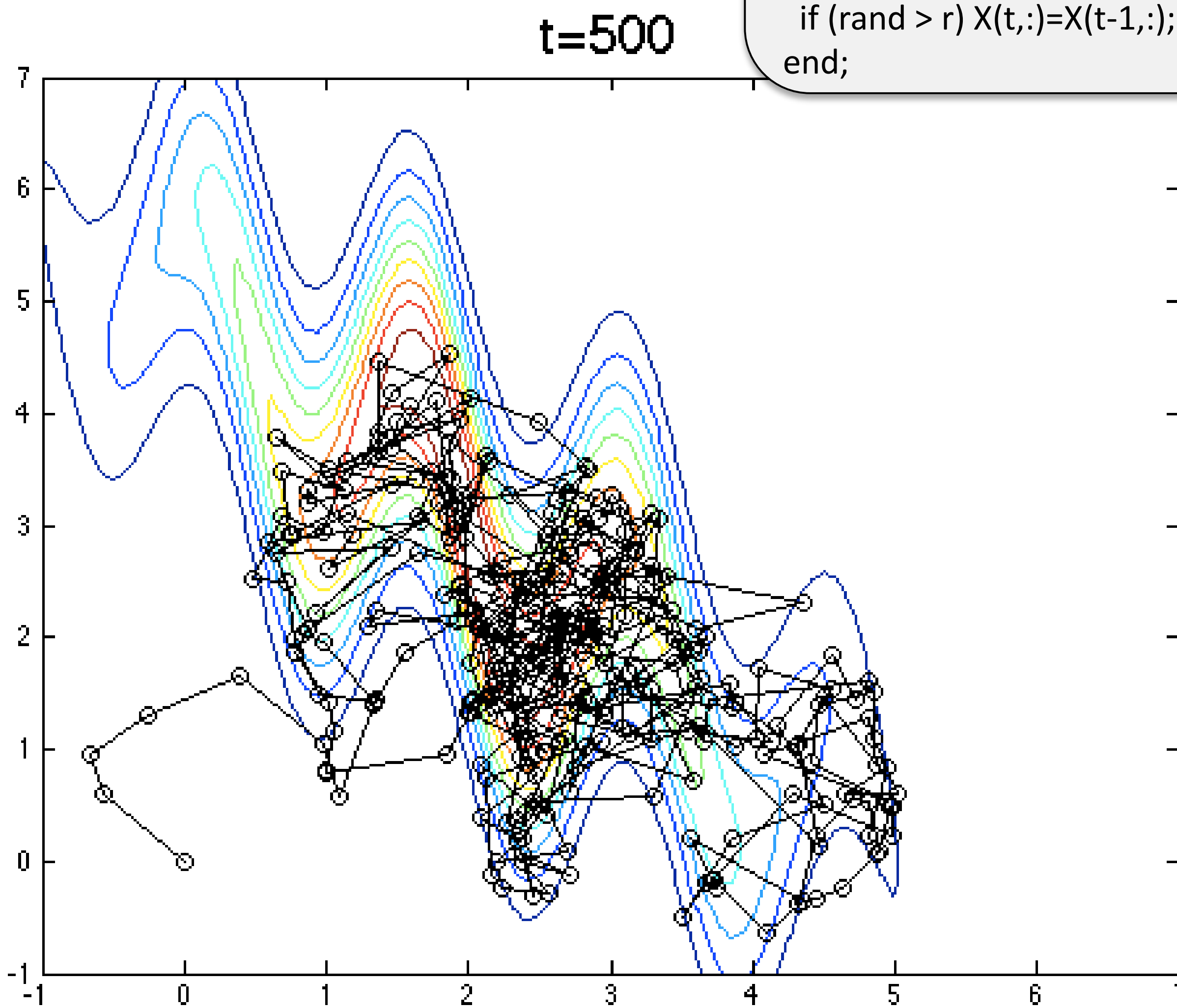
```
f = @(X) ...                          % define f(x) / p(x), target
X = [0,0];                            % set or sample initial state
for t=2:T,                            % simulate Markov chain:
  X(t,:) = X(t-1,:) + .5*randn(1,2);     % propose move
  r = min( 1, f(X(t,:)) / f(X(t-1,:)) );  % compute acceptance
  if (rand > r) X(t,:)=X(t-1,:); end;    % sample acceptance
end;
```



MCMC Example

Metropolis-Hastings (symmetric proposal)

```
f = @(X) ...                              % define f(x) / p(x), target
X = [0,0];                                % set or sample initial state
for t=2:T,                                % simulate Markov chain:
  X(t,:) = X(t-1,:) + .5*randn(1,2);     % propose move
  r = min( 1, f(X(t,:)) / f(X(t-1,:)) ); % compute acceptance
  if (rand > r) X(t,:)=X(t-1,:); end;    % sample acceptance
end;
```



Asymptotically, samples will represent p(x)

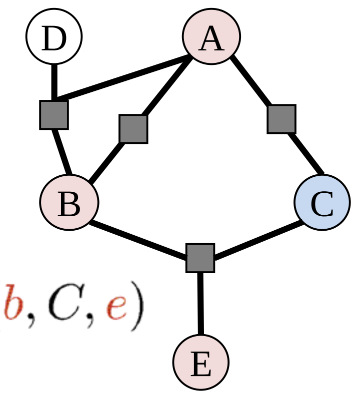
Gibbs sampling

[Geman & Geman 1984]

- Proceed in rounds
  - Sample each variable in turn given all the others' most recent values:

$$x'_0 \sim p(X_0|x_1, x_2, x_3)$$
$$x'_1 \sim p(X_1|x'_0, x_2, x_3)$$
$$x'_2 \sim p(X_2|x'_0, x'_1, x_3)$$
$$\vdots$$

$$c \sim p(C \mid \ldots)$$
$$\propto f(a, C)\, f(b, C, e)$$



  - Conditional distributions depend only on the Markov blanket
  - Easy to see that p(x) is a stationary distribution:

$$\sum_{x_1} p(x'_1|x_2 \ldots x_n) p(x_1, \ldots x_n) = p(x'_1|x_2 \ldots x_n) p(x_2, \ldots x_n) = p(x'_1, x_2 \ldots x_n)$$

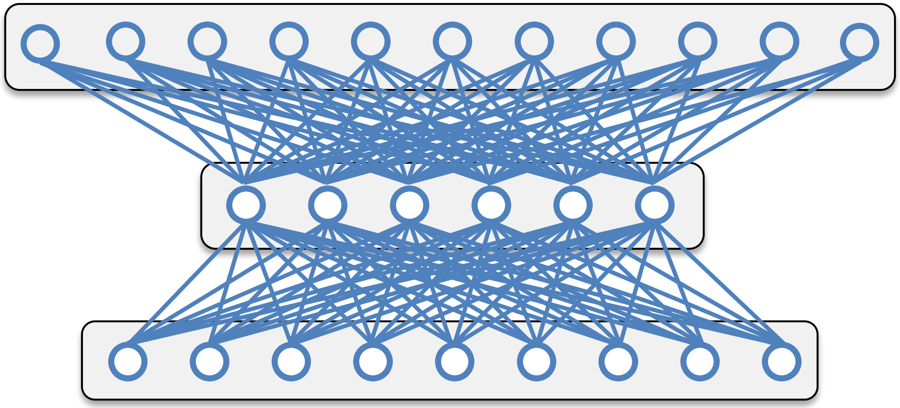
Advantages:
No rejections
No free parameters (q)

Disadvantages:
"Local" moves
May mix slowly if vars strongly correlated
(can fail with determinism)

Ex: DBMs

- Very popular for restricted / deep Boltzmann machines
  - Each layer is independent given surrounding layers

- Used in both
  - model training (estimate gradient of LL)
    - Contrastive divergence; persistent CD; …
  - model validation (estimate log-likelihood of data)
    - Annealed & reverse annealed importance sampling; discriminance sampling



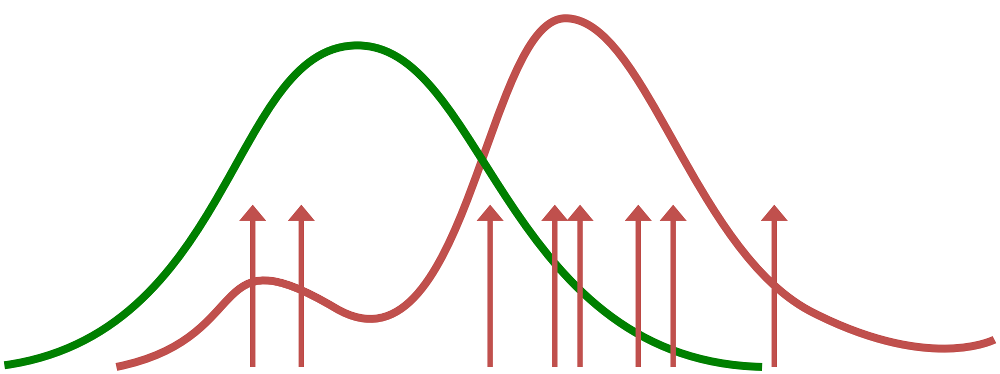
MCMC and Common Queries

- MCMC generates samples (asymptotically) from p(x)
- Estimating expectations is straightforward

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \{x^{(i)}\} \sim p(x)$$

- Estimating the partition function

$$\frac{1}{Z} = \int_x p_0(x)\,\frac{1}{Z} = \int_x p_0(x)\,\frac{p(x)}{f(x)}$$



MCMC and Common Queries

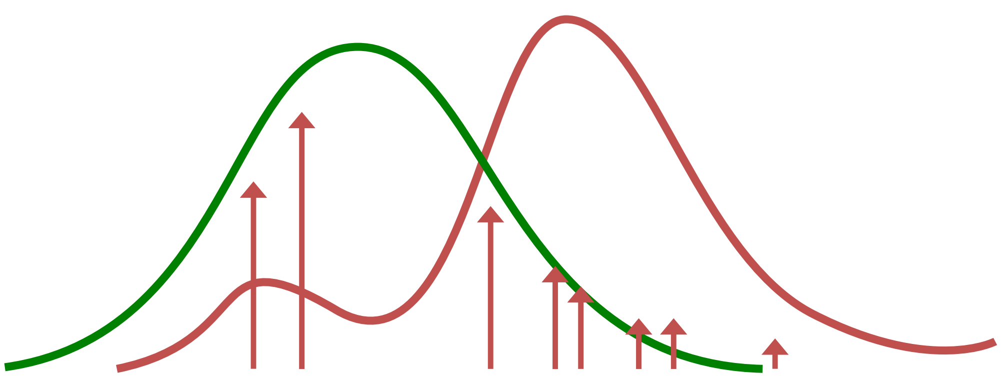
- MCMC generates samples (asymptotically) from p(x)

- Estimating expectations is straightforward

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \{x^{(i)}\} \sim p(x)$$

- Estimating the partition function

$$\frac{1}{Z} = \int_x p_0(x)\,\frac{1}{Z} = \int_x p_0(x)\,\frac{p(x)}{f(x)} \approx \frac{1}{n}\sum_i \frac{p_0(x^{(i)})}{f(x^{(i)})}$$



"Reverse" importance sampling

$$\hat{Z}_{ris} = \left[\frac{1}{n}\sum_i \frac{p_0(x^{(i)})}{f(x^{(i)})}\right]^{-1}$$

Ex: Harmonic Mean Estimator
[Newton & Raftery 1994; Gelfand & Dey, 1994]

$$f(x) = p(D|\theta)p(\theta) \qquad p_0(x) = p(\theta)$$

MCMC

- Samples from p(x) asymptotically (in time)
  - Samples are not independent
- Rate of convergence ("mixing") depends on
  - Proposal distribution for MH
  - Variable dependence for Gibbs
- Good choices are critical to getting decent performance
- Difficult to measure mixing rate; lots of work on this

- Usually discard initial samples ("burn in")
  - Not necessary in theory, but helps in practice
- Average over rest; asymptotically unbiased estimator

$$\mathbb{E}[u(x)] = \int p(x)u(x) \quad \approx \quad \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

Monte Carlo

Importance sampling

- i.i.d. samples
- Unbiased estimator
- Bounded weights provide finite-sample guarantees

- Samples from Q
- Good proposal: close to p but easy to sample from

- Reject samples with zero-weight

MCMC sampling

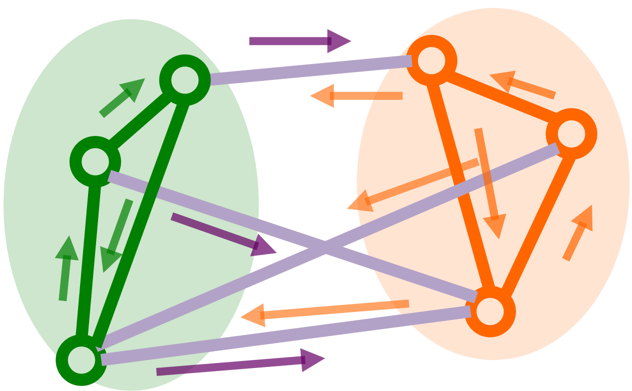
- Dependent samples
- Asymptotically unbiased
- Difficult to provide finite-sample guarantees

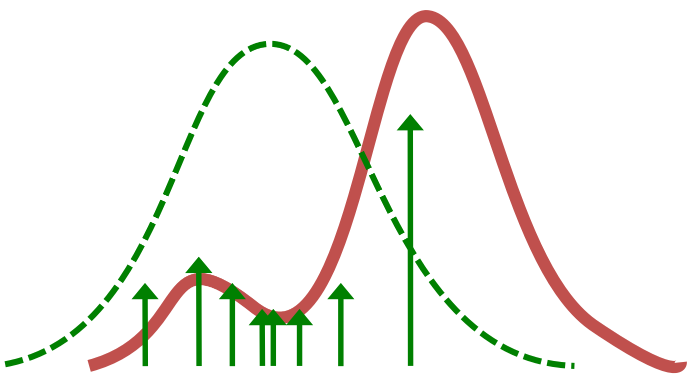
- Samples from ¼ P(X|e)
- Good proposal: move quickly among high-probability x

- May not converge with deterministic constraints

Outline

- Review: Graphical Models
- Variational methods
  - Convexity & decomposition bounds
  - Variational forms & the marginal polytope
  - Message passing algorithms
  - Convex duality relationships
- **Monte Carlo sampling**
  - Basics
  - Importance sampling
  - Markov chain Monte Carlo
  - **Integrating inference and sampling**





Estimating with samples

- Suppose we want to estimate $p(X_i \mid E)$

- Method 1: histogram (count samples where $X_i=x_i$)

$$P(X_i = x_i|E) \approx \frac{1}{m}\sum_t \mathbb{1}[\tilde{x}_i^{(t)} = x_i] \qquad \tilde{x}^{(t)} \sim p(X|E)$$

- Method 2: average probabilities

$$P(X_i = x_i|E) \approx \frac{1}{m}\sum_t p(x_i \,|\, \tilde{x}_{\neg i}^{(t)}) \qquad \tilde{x}^{(t)} \sim p(X|E)$$

Converges faster! (uses all samples)

Rao-Blackwell Theorem: [e.g., Liu et al. 1995]

Let $X = (X_S,X_T)$, with joint distribution $p(X_S,X_T)$, to estimate $\mathbb{E}[u(X_S)]$

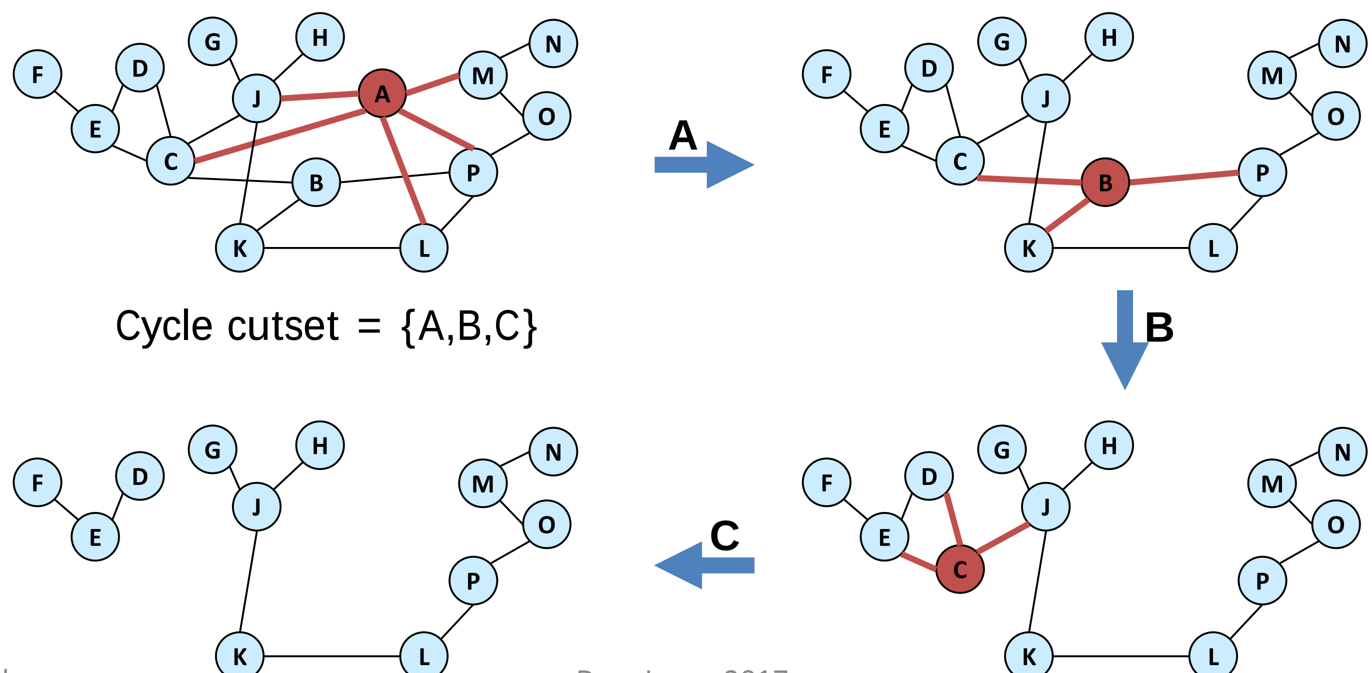
Then, $\mathrm{Var}\Big[\mathbb{E}[u(X_S)|X_T]\Big] \;\leq\; \mathrm{Var}\Big[u(X_S)\Big]$

Weak statement, but powerful in practice! Improvement depends on X_S,X_T

Cutsets

- Exact inference:
  - Computation is exponential in the graph's induced width

- "w-cutset": set C, such that $p(X_{:C} \mid X_C)$ has induced width w
  - "cycle cutset": resulting graph is a tree; w=1

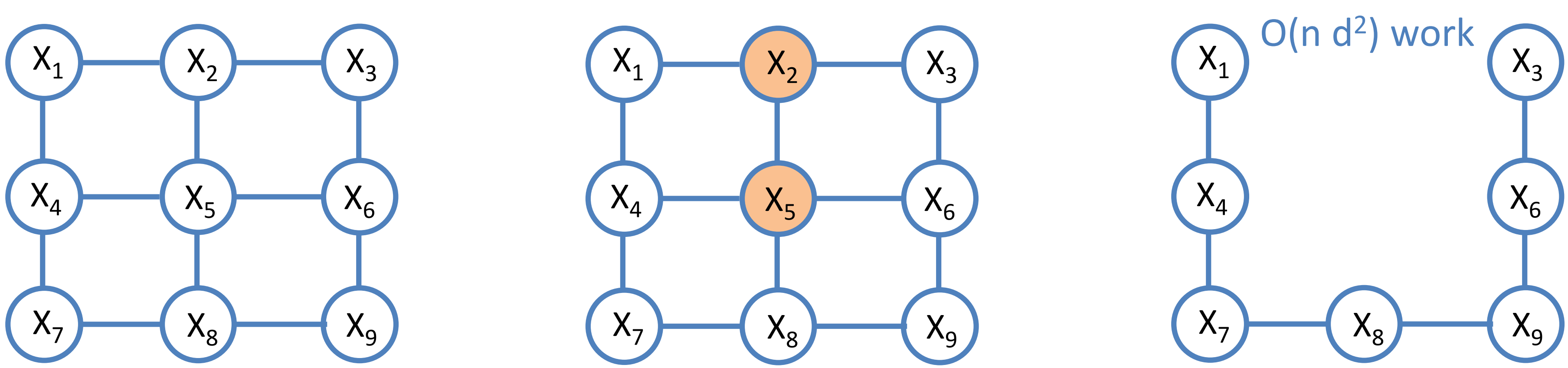


Cutset Importance Sampling

[Gogate & Dechter 2005, Bidyuk & Dechter 2006]

- Use cutsets to improve estimator variance
  - Draw a sample for a w-cutset X_C
  - Given X_C, inference is O(exp(w))

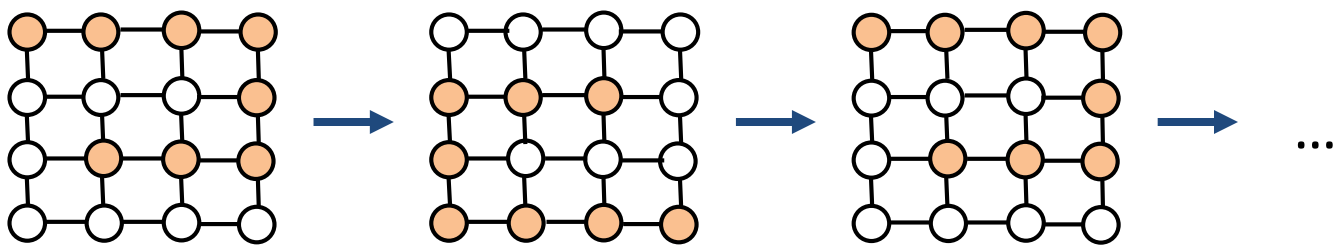
$F(X) = \prod f_{ij}(X_i, X_j)$ $\quad$ $\tilde{x}_2^{(i)}, \tilde{x}_5^{(i)} \sim q(X_2, X_5)$ $\quad$ $F(\tilde{x}_2^{(i)}, \tilde{x}_5^{(i)})$



(Use weighted sample average for X_C; weighted average of probabilities for $X_{\neg C}$)

Using Inference in Gibbs sampling

- "Blocked" Gibbs sampler
  - Sample several variables together

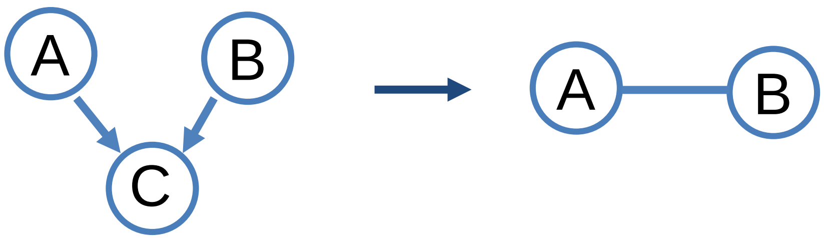


$x_A^{(1)} \sim p(X_A | x_B^{(0)})$ $x_B^{(1)} \sim p(X_B | x_A^{(1)})$ $x_A^{(2)} \sim p(X_A | x_B^{(1)})$

  - Cost of sampling is exponential in the block's induced width
  - Can significantly improve convergence (mixing rate)
  - Sample strongly correlated variables together

Using Inference in Gibbs sampling

- "Collapsed" Gibbs sampler
  - Analytically marginalize some variables before / during sampling

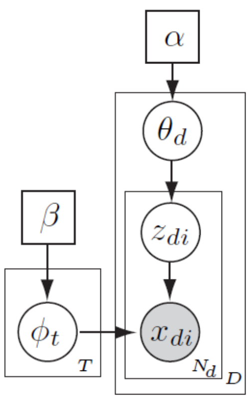


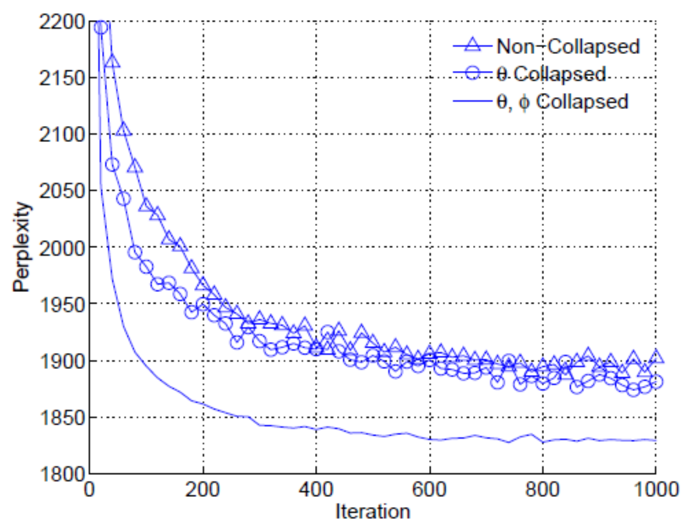
$$a^{(1)} \sim \sum_c p(A, c \,|\, b^{(0)})$$

$$b^{(1)} \sim \sum_c p(B, c \,|\, a^{(1)})$$

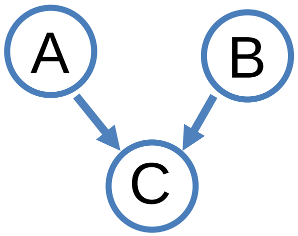
$$\vdots$$

  - Ex: LDA "topic model" for text





Using Inference in Gibbs Sampling



Faster Convergence

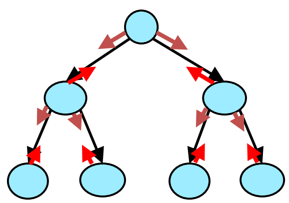
- Standard Gibbs:
  $$p(A \mid b, c) \rightarrow P(B \mid a, c) \rightarrow P(C \mid a, b) \qquad (1)$$
- Blocking:
  $$p(A \mid b, c) \rightarrow P(B, C \mid a) \qquad (2)$$
- Collapsed:
  $$p(A \mid b) \rightarrow P(B \mid a) \qquad (3)$$

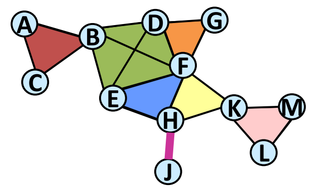
Summary: Monte Carlo methods

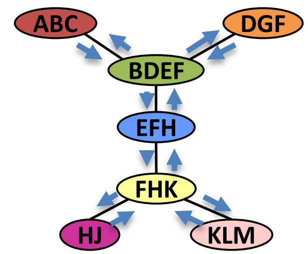
- Stochastic estimates based on sampling
  - Asymptotically exact, but few guarantees in the short term
- Importance sampling
  - Fast, potentially unbiased
  - Performance depends on a good choice of proposal q
  - Bounded weights can give finite sample, probabilistic bounds
- MCMC
  - Only asymptotically unbiased
  - Performance depends on a good choice of transition distribution
- Incorporating inference
  - Use exact inference within sampling
  - Reduces the variance of the estimates

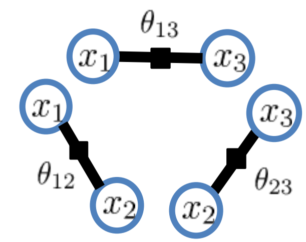
Course Summary

- Class 1: Introduction and Inference

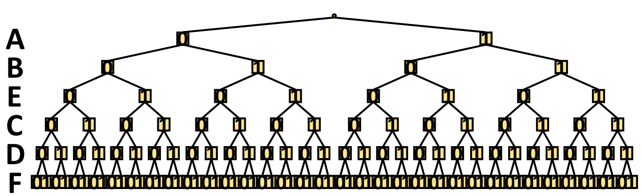


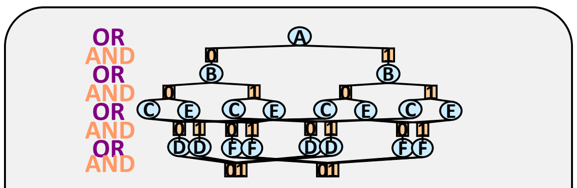






- Class 2: Search





Context minimal AND/OR search graph

- Class 3: Variational Methods and Monte Carlo Sampling