Probabilistic Graphical Models

Lecture 10: Variational Inference

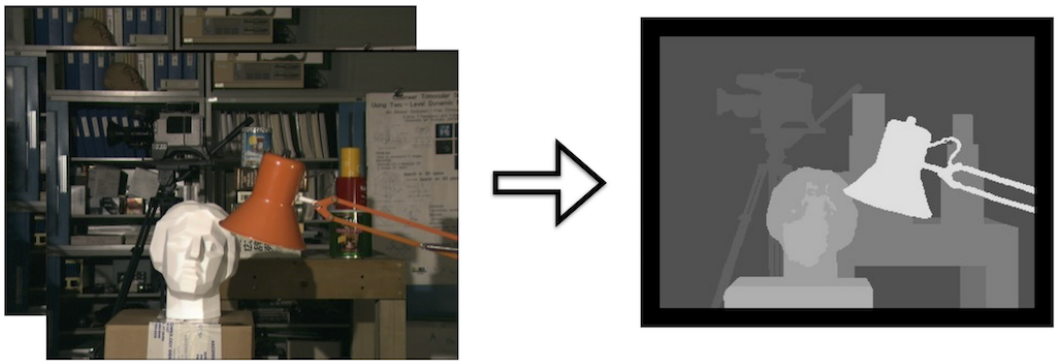
Matthew Walter

TTI-Chicago

May 7, 2020

Approximate Inference

- Given a graphical model over $\boldsymbol{X}$ and evidence $\boldsymbol{E} = \boldsymbol{e}$, we are interested in conditional probability queries $P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e})$ for $\boldsymbol{Y} \subseteq \boldsymbol{X}$
- While exact inference is NP-hard, several real-world inference problems are easy (e.g., hidden Markov models)
- However, exact inference is intractable for many problems
- For example, dense stereo reconstruction: For every pixel (i_l, j_l) in left image, we are interested in the image-space distance (**disparity**) y_{i_l, j_l} to the corresponding pixel (i_r, j_r) in the right image



Approximate Inference

- Given a graphical model over $\boldsymbol{X}$ and evidence $\boldsymbol{E} = \boldsymbol{e}$, we are interested in conditional probability queries $P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e})$ for $\boldsymbol{Y} \subseteq \boldsymbol{X}$
- While exact inference is NP-hard, several real-world inference problems are easy (e.g., hidden Markov models)
- However, exact inference is intractable for many problems
- *Approximate inference* provides a tractable alternative
- Nearly all approximate algorithms are either:
  1. Variational algorithms (e.g., mean-field, loopy belief propagation)
  2. Monte-carlo methods (e.g., MCMC)
- The next two lectures will focus on variational methods

Variational Methods

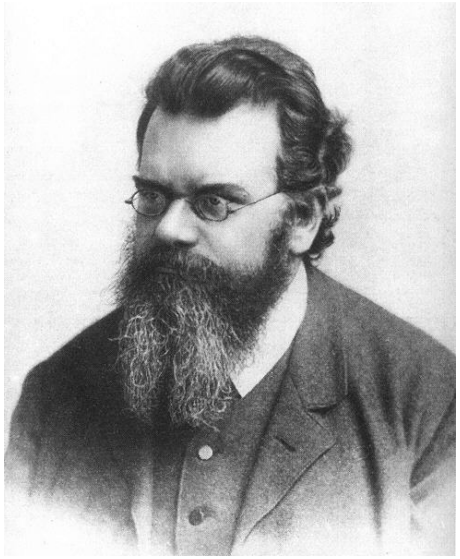
- **Goal**: Approximate a difficult distribution $P(\boldsymbol{X} \mid \boldsymbol{e})$ with a new distribution $Q(\boldsymbol{X})$ such that:
  1. $P(\boldsymbol{X} \mid \boldsymbol{e})$ and $Q(\boldsymbol{X})$ are "close"
  2. Inference on $Q(\boldsymbol{X})$ is easy
- How should we measure the distance between distributions?

Entropy

- Per Boltzman, the number of microstates of a system in thermodynamic equilibrium (i.e., the uncertainty in a system)

$$\text{entropy} = k_B \ln \Omega$$

where k_B is the Boltzmann constant and Ω is the number of microstates



Ludwig Boltzmann

(1844–1906)

- Gibbs entropy: $-k_B \sum_i p_i \ln p_i$ where $p_i = P(\text{microstate})$
- These are equivalent when microstates are equally likely (i.e., $P = \frac{1}{\Omega}$)

Entropy

- Measure of the amount of "noise" in a distribution
- Expected amount of information in an event drawn from the distribution
- Alternatively, a measure of the number of bits needed on average to encode an outcome

$$H_P(P(\boldsymbol{X})) = -\sum_{\boldsymbol{x} \in \text{Val}(\boldsymbol{X})} P(\boldsymbol{x}) \ln P(\boldsymbol{x})$$
$$= \mathbb{E}_P[-\ln P(\boldsymbol{x})]$$

(where this assumes a discrete distribution)

- Also known as the "Shannon entropy"

Entropy of a Markov Network

- Consider a Markov network over $\boldsymbol{X}$ that factorizes as

$$P(\boldsymbol{X}) = \frac{1}{Z} \prod_k \phi_k(\boldsymbol{D}_k)$$

Entropy of a Markov Network

- Consider a Markov network over $\boldsymbol{X}$ that factorizes as

$$P(\boldsymbol{X}) = \frac{1}{Z} \prod_k \phi_k(\boldsymbol{D}_k)$$

- The entropy is then

$$H_P(\boldsymbol{X}) = \ln Z + \sum_k \mathbb{E}_P \left[-\ln \phi_k(\boldsymbol{D}_k) \right]$$

Entropy of a Markov Network

- Consider a Markov network over $\boldsymbol{X}$ that factorizes as

$$P(\boldsymbol{X}) = \frac{1}{Z} \prod_k \phi_k(\boldsymbol{D}_k)$$

- The entropy is then

$$H_P(\boldsymbol{X}) = \ln Z + \sum_k \mathbb{E}_P \left[-\ln \phi_k(\boldsymbol{D}_k) \right]$$

- Number of summations is a function of the number of factors and their local scope (compared to number of joint assignments to $\boldsymbol{X}$)

Entropy of a Markov Network

- Consider a Markov network over $\boldsymbol{X}$ that factorizes as

$$P(\boldsymbol{X}) = \frac{1}{Z} \prod_k \phi_k(\boldsymbol{D}_k)$$

- The entropy is then

$$H_P(\boldsymbol{X}) = \ln Z + \sum_k \mathbb{E}_P \left[-\ln \phi_k(\boldsymbol{D}_k) \right]$$

- Number of summations is a function of the number of factors and their local scope (compared to number of joint assignments to $\boldsymbol{X}$)
- Still requires computing Z and marginal distribution over the scope of each factor $\boldsymbol{D}_k$

Entropy of a Bayesian Network

- Consider a Bayesian network over $\boldsymbol{X}$ that factorizes as

$$P(\boldsymbol{X}) = \prod_i P(X_i \,|\, \mathsf{Pa}_{X_i}^G)$$

- The entropy decomposes as

$$H_P(\boldsymbol{X}) = \sum_i H_P(X_i \,|\, \mathsf{Pa}_i^G)$$

- CPDs can be used to establish bounds on the entropy

A Crash Course on Information Theory

- Consider a discrete random variable X with distribution $P(X)$
- Suppose we want to transmit $x \sim P(X)$ in a lossless fashion

A Crash Course on Information Theory

- Consider a discrete random variable X with distribution $P(X)$
- Suppose we want to transmit $x \sim P(X)$ in a lossless fashion
- Let $C : \text{Val}(X) \rightarrow \{0, 1\}^+$ be an **extended code**
- Let $c(x)$ be the **codeword** for $x \in \text{Val}(X)$ with length $l(x)$
- We want a code that is:
  - Uniquely decodeable: $\forall a, b \in \text{Val}(X), a \neq b \Rightarrow c^+(a) \neq c^+(b)$
  - Easy to decode: No codeword can be a prefix for another codeword

$$C_1 = \{0, 101\} \text{ vs. } C_2 = \{1, 101\}$$

  - Code should achieve as much compression as possible
- **Prefix codes** are uniquely decodeable and easy to decode

A Crash Course on Information Theory

- The **expected length** of a code C for random variable X is

$$L(C, X) = \mathbb{E}_P[l(X)] = \sum_x l(x)P(x)$$

- Example: Let $\text{Val}(X) = \{a, b, c, d\}$ and $P(X) = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$
  - Consider fixed-length code $C_1 = \{00, 01, 10, 11\}$:

$$L(C_1, X) = 2 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + 2 \cdot \frac{1}{8} = 2$$

A Crash Course on Information Theory

- The **expected length** of a code C for random variable X is

$$L(C, X) = \mathbb{E}_P[l(X)] = \sum_x l(x)P(x)$$

- Example: Let $\text{Val}(X) = \{a, b, c, d\}$ and $P(X) = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$
  - Consider fixed-length code $C_1 = \{00, 01, 10, 11\}$:

$$L(C_1, X) = 2 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + 2 \cdot \frac{1}{8} = 2$$

  - Consider variable-length code $C_2 = \{0, 10, 110, 111\}$:

$$L(C_2, X) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = 1.75$$

(we can save bits by shortening $00 \rightarrow 0$, but have to lengthen other codewords to retain unique decodeability)

A Crash Course on Information Theory

- We want to minimize expected code length $L(C, X) = \sum_x l(x)P(x)$
- **Source coding theorem** (Shannon, 1948):

$$L(C, X) \geq H(X)$$

- Optimal codelengths: $l(x) = \log_2 \frac{1}{P(x)}$ (Shannon information content)
- Example: $\text{Val}(X) = \{a, b, c, d\}$ and $P(X) = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$

$$C_2 = \{0, 10, 110, 111\} \text{ is the optimal code}$$

KL Divergence (Relative Entropy)

- The **Kullback-Liebler divergence** (KL-divergence, "relative entropy") between two distributions P and Q is defined as

$$D(P\|Q) = \mathbb{E}_P\left[\log \frac{P(\boldsymbol{X})}{Q(\boldsymbol{X})}\right] \underset{\text{discrete}}{=} \sum_{\boldsymbol{x}\in \text{Val}(\boldsymbol{X})} P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}$$

- **Gibbs' Inequality**: $D(P\|Q) \geq 0$ for all P and Q and zero iff $P = Q$
- Measure of the "distance" between two distributions
- KL-divergence is **not symmetric**, i.e., $D(P\|Q) \neq D(Q\|P)$
- KL-divergence does not obey the triangle inequality
- Thus, KL-divergence is thus not a metric

KL Divergence (Relative Entropy)

$$D(P\|Q) = \mathbb{E}_P\left[\ln \frac{P(\boldsymbol{X})}{Q(\boldsymbol{X})}\right] \underset{\text{discrete}}{=} \sum_{\boldsymbol{x}\in \text{Val}(\boldsymbol{X})} P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}$$

- Consider a discrete random variable X with distribution $P(X)$
- Suppose we use a complete code with length $l(X)$, which define implicit probabilities $Q(X) = 2^{-l(X)}$
- The average length is

$$\begin{aligned} L(C, X) &= H(X) + \sum_x P(X) \log \frac{P(x)}{Q(X)} \\ &= H(X) + D(P\|Q) \end{aligned}$$

- KL-divergence measures the cost of using the wrong codelengths

KL-Divergence

$$D(P\|Q) = \sum_{\boldsymbol{x}} P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}$$

- Let P be the true distribution that we want to perform inference over
- Due to asymmetry, there are two options for the optimization:
  - **M-projection** (Moment projection) of Q onto P

$$Q_M^* = \arg\min_Q D(P\|Q)$$

  - **I-projection** (Information projection) of Q onto P

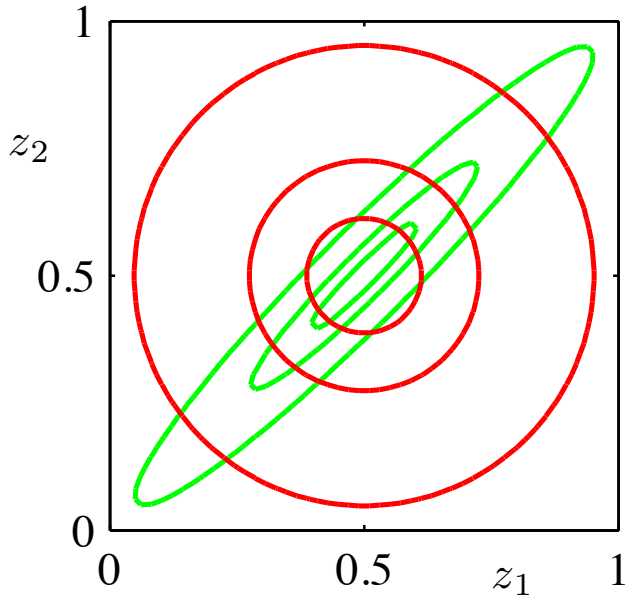
$$Q_I^* = \arg\min_Q D(Q\|P)$$

- These two will differ when Q is minimized over a restricted set of distributions, i.e., $\mathcal{Q} = \{Q_1, \ldots, Q_n\}$, where $P \notin \mathcal{Q}$

KL-Divergence: M-Projection

$$q_M^*(\boldsymbol{x}) = \arg\min_{q \in \mathcal{Q}} D(p\|q) = \arg\min_{q \in \mathcal{Q}} \int_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$$

Suppose that $p(\boldsymbol{x})$ is a 2D Gaussian and that $\mathcal{Q}$ is the set of all 2D Gaussians with diagonal covariance matrices

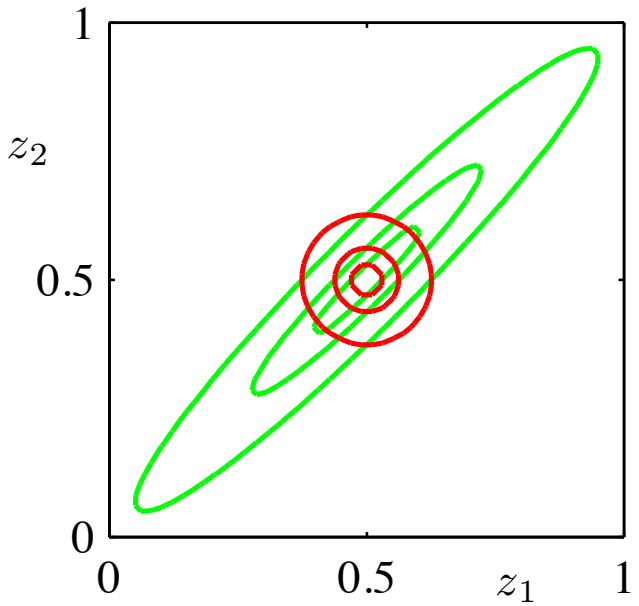


Green: $p(\boldsymbol{x})$, Red: $q_M^*(\boldsymbol{x})$

KL-Divergence: I-Projection

$$q_I^*(\boldsymbol{x}) = \arg\min_{q \in \mathcal{Q}} D(q\|p) = \arg\min_{q \in \mathcal{Q}} \int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$

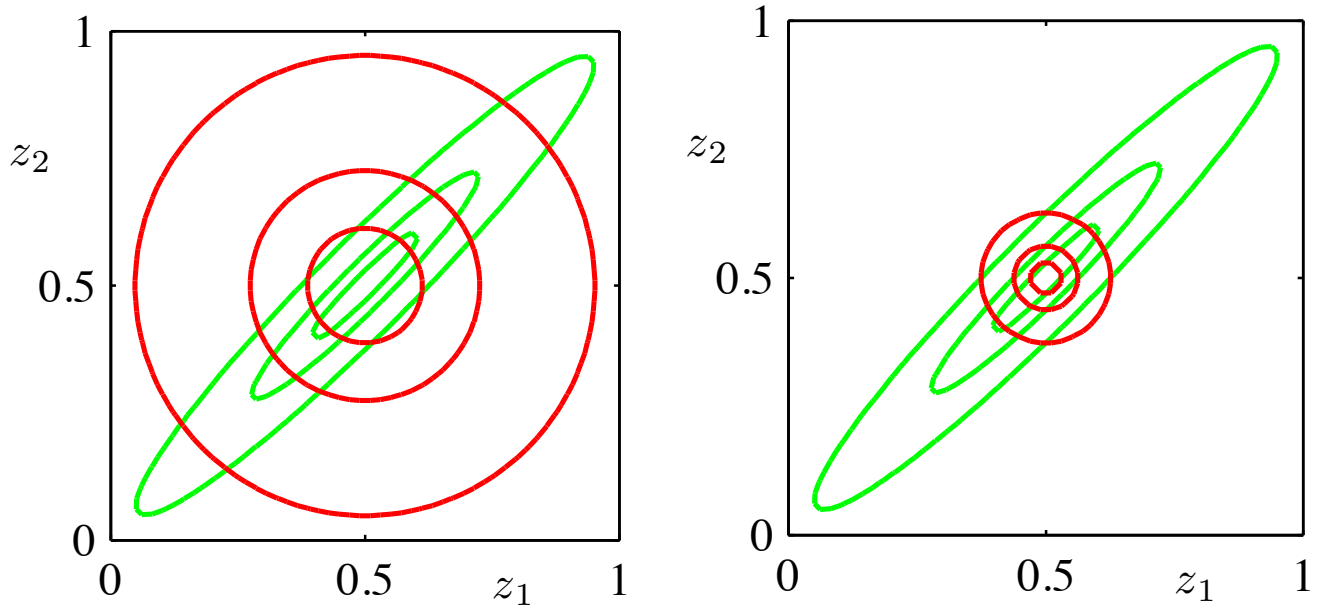
Suppose that $p(\boldsymbol{x})$ is a 2D Gaussian and that $\mathcal{Q}$ is the set of all 2D Gaussians with diagonal covariance matrices



Green: $p(\boldsymbol{x})$, Red: $q_I^*(\boldsymbol{x})$

KL-Divergence: Single Gaussian

In this unimodal Gaussian example, both the M-projection and I-projection yield an approximate distribution with the correct mean.



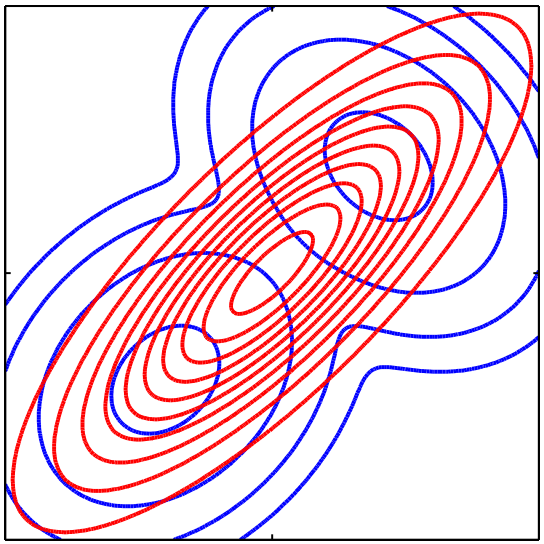
Green: $p(\boldsymbol{x})$, Red: $q^*(\boldsymbol{x})$

But, what if $p(\boldsymbol{x})$ is multimodal?

KL-Divergence: M-Projection (Mixture of Gaussians)

$$q_M^*(\boldsymbol{x}) = \arg\min_{q \in \mathcal{Q}} D(p\|q) = \arg\min_{q \in \mathcal{Q}} \int_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$$

Suppose that $p(\boldsymbol{x})$ is a mixture of 2D Gaussians and that $\mathcal{Q}$ is the set of all 2D Gaussians (with arbitrary covariance matrices)



Blue: $p(\boldsymbol{x})$, Red: $q_M^*(\boldsymbol{x})$

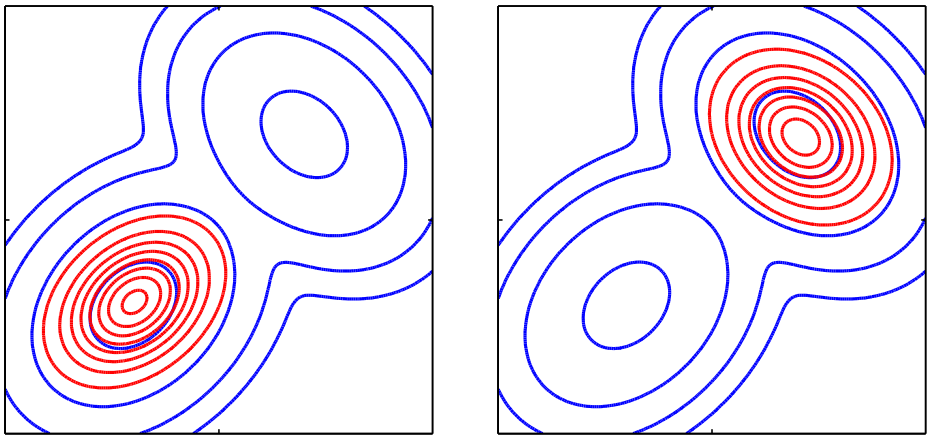
M-projection yields distribution $q(\boldsymbol{x})$ with the correct mean and covariance

KL-Divergence: I-Projection (Mixture of Gaussians)

$$q_I^*(\boldsymbol{x}) = \arg\min_{q \in \mathcal{Q}} D(q\|p) = \arg\min_{q \in \mathcal{Q}} \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$

Suppose that $p(\boldsymbol{x})$ is a 2D Gaussian and that $\mathcal{Q}$ is the set of all 2D Gaussians with diagonal covariance matrices

Two equally good solutions



Blue: $p(\boldsymbol{x})$, Red: $q_I^*(\boldsymbol{x})$ (two local minima)

Unlike M-projection, I-projection may not yield correct moments

KL-Divergence: M-Projection and Exact Inference

$$Q_M^*(\boldsymbol{X}) = \arg\min_{Q \in \mathcal{Q}} D(P \| Q) = \arg\min_{Q \in \mathcal{Q}} \sum_{\boldsymbol{x}} P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}$$

- Consider M-projection where we restrict $\mathcal{Q}$ to be an exponential family
- **Theorem**: The expected sufficient statistics w.r.t. Q_M^* are exactly the marginals of P (i.e., M-projection performs moment matching)

$$\mathbb{E}_{Q^*}[\tau(\boldsymbol{X})] = \mathbb{E}_P[\tau(\boldsymbol{X})]$$

  where $\tau(\boldsymbol{x})$ are the *sufficient statistics*
- Computing the M-projection requires computing the marginals of P
- M-projection requires expectation w.r.t. P and is thus has hard as exact inference
- Variational inference typically operates using the I-projection (requires expectation w.r.t. Q, but that is much easier by design)

Variational Methods

- Suppose that we have a general graphical model

$$P(\boldsymbol{X};\theta) = \frac{1}{Z(\theta)} \prod_{c \in \mathcal{C}} \phi_c(\boldsymbol{X}_c) = \exp\left(\sum_{c \in \mathcal{C}} \theta_c(\boldsymbol{X}_c) - \ln Z(\theta) \right)$$

- The methods begin as follows

$$\begin{aligned} D(Q\|P) &= \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \frac{Q(\boldsymbol{X})}{P(\boldsymbol{X})} \\ &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln P(\boldsymbol{X}) - \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \frac{1}{Q(\boldsymbol{X})} \end{aligned}$$

Variational Methods

- Suppose that we have a general graphical model

$$P(\boldsymbol{X};\theta) = \frac{1}{Z(\theta)} \prod_{c \in \mathcal{C}} \phi_c(\boldsymbol{X}_c) = \exp\left(\sum_{c \in \mathcal{C}} \theta_c(\boldsymbol{X}_c) - \ln Z(\theta) \right)$$

- The methods begin as follows

$$\begin{aligned} D(Q\|P) &= \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \frac{Q(\boldsymbol{X})}{P(\boldsymbol{X})} \\ &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln P(\boldsymbol{X}) - \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \frac{1}{Q(\boldsymbol{X})} \\ &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \left(\sum_{c \in \mathcal{C}} \theta_c(\boldsymbol{X}_c) - \ln Z(\theta) \right) - H(Q(\boldsymbol{X})) \\ &= -\sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \theta_c(\boldsymbol{X}_c) + \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Z(\theta) - H(Q(\boldsymbol{X})) \\ &= -\sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + \ln Z(\theta) - H(Q(\boldsymbol{X})) \end{aligned}$$

Variational Methods

$$D(Q\|P) = -\left\{\sum_{c\in\mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))\right\} + \ln Z(\theta)$$
$$= -F[\tilde{P}, Q] + \ln Z(\theta)$$

- In physics, $F[\tilde{P}, Q]$ is called the (negative) **variational free energy** (alt: **Helmhotz free energy** or **variational lower bound**), where
  - *Energy term*: $\sum_{c\in\mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] = \mathbb{E}_Q\left[\sum_{c\in\mathcal{C}} \ln \phi_c(\boldsymbol{X}_c)\right]$
  - *Entropy term* is the entropy over Q
- Computing the energy term involves expectation of individual factors
- The complexity of computing both terms is a function of Q (not P)
- We can force Q to be closer to P by maximizing the energy functional

Variational Methods: Optimizing the Energy Functional

- Since $D(Q\|P) \geq 0$, we have

$$-\sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + \ln Z(\theta) - H(Q(\boldsymbol{X})) \geq 0,$$

  which implies that

$$\ln Z(\theta) \geq \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- A good approximation (i.e., $D(Q\|P)$ is small) provides a tighter lower-bound on the log-partition function (for a BN, this is the probability of evidence)
- Since $D(Q\|P) = 0$ only when $P = Q$, if we are able to optimize over *all* distributions, we have

$$\ln Z(\theta) = \max_Q \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

Variational Methods: Optimizing the Energy Functional

- Consider approximating $P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e})$ with $Q(\boldsymbol{Y})$[1]
- Expanding the expression for the KL-divergence and using Bayes' rule

$$\begin{aligned} D(Q\|P) &= \sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log \frac{Q(\boldsymbol{y})P(\boldsymbol{e})}{P(\boldsymbol{y} \mid \boldsymbol{e})} \\ &= -\sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log Q(\boldsymbol{y}) - \sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log P(\boldsymbol{y}, \boldsymbol{e}) + \log P(\boldsymbol{e}) \end{aligned}$$

- Rearranging terms, we have

$$\begin{aligned} \log P(\boldsymbol{e}) &= \left(\sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log Q(\boldsymbol{y}) + \sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log P(\boldsymbol{y}, \boldsymbol{e}) \right) + D(Q\|P) \\ &= F[P, Q] + D(Q\|P) \end{aligned}$$

- $F[P, Q]$ is the **variational** (or **evidence**) **lower bound** (or (ELBO))

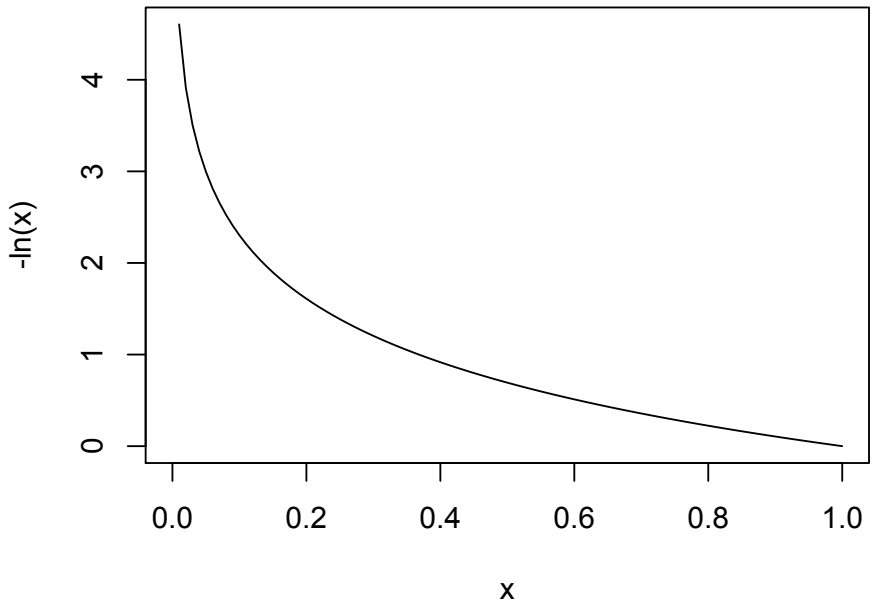
[1] Since $\boldsymbol{E} = \boldsymbol{e}$ is observed, we treat it as a parameter of Q.

Variational Methods: Tangent

- For any λ and any x:

$$-\ln x \geq -\lambda x + \ln \lambda + 1$$

where the right-hand side is a family of functions $g_\lambda(x)$

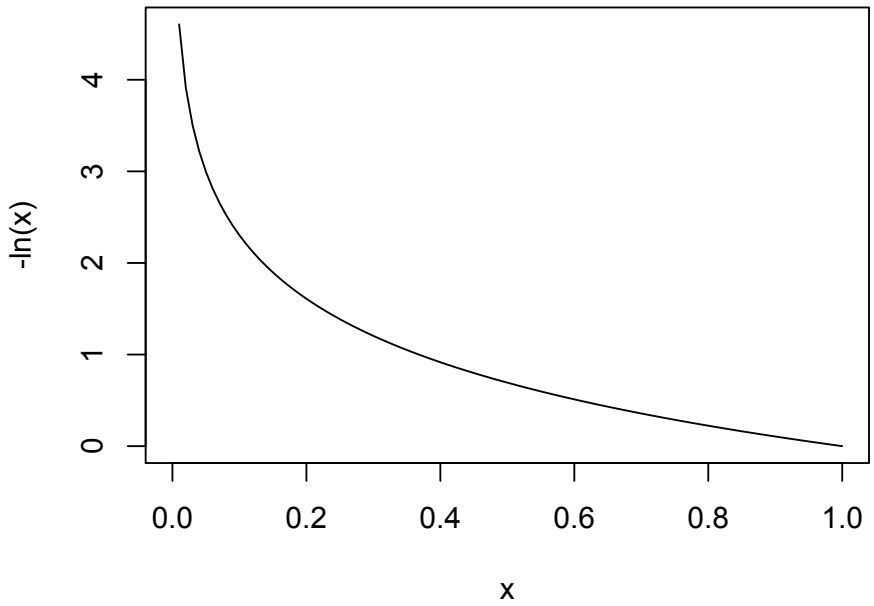


Variational Methods: Tangent

- For any λ and any x:

$$-\ln x \geq -\lambda x + \ln \lambda + 1$$

where the right-hand side is a family of functions $g_\lambda(x)$



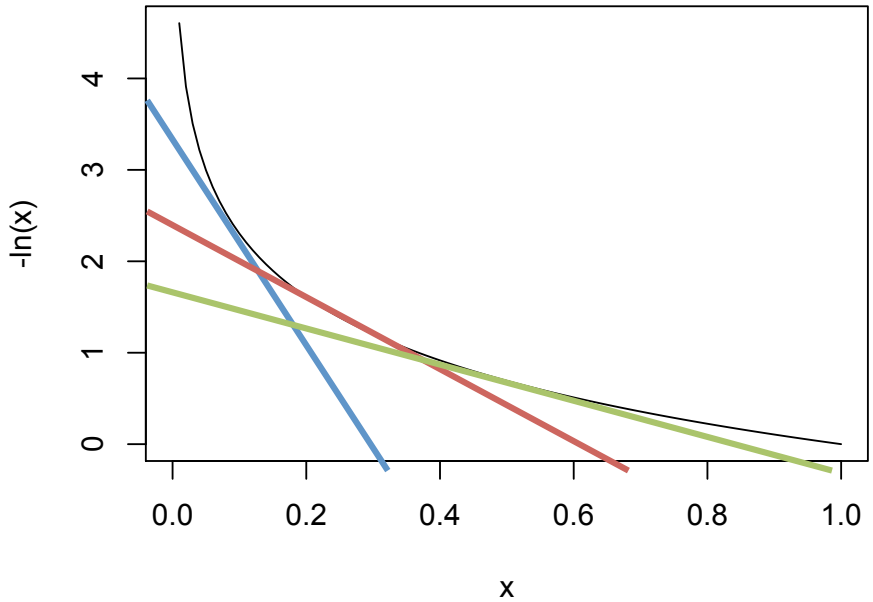
- For any x, there is some λ for which the bound is tight
  - λ is called a **variational parameter**

Variational Methods: Tangent

- For any λ and any x:

$$-\ln x \geq -\lambda x + \ln \lambda + 1$$

where the right-hand side is a family of functions $g_\lambda(x)$



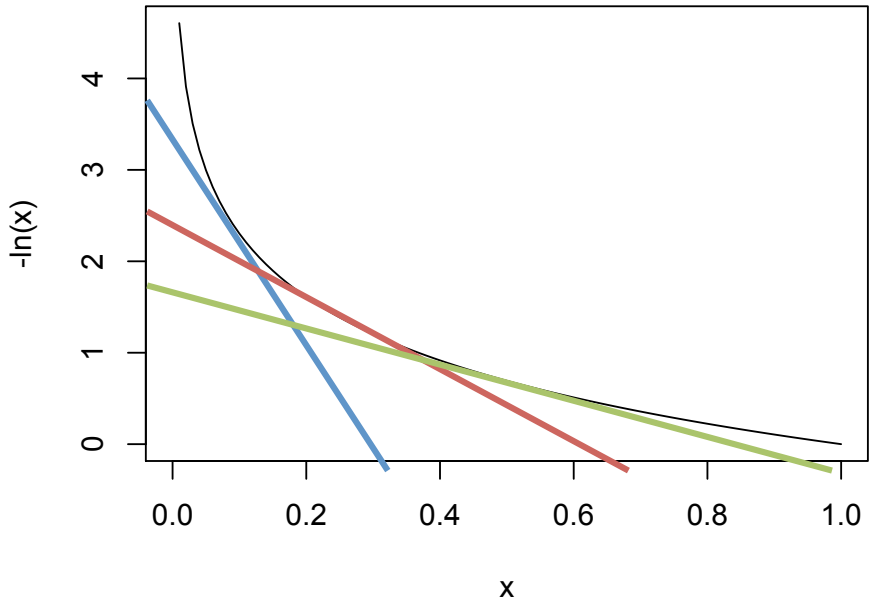
- For any x, there is some λ for which the bound is tight
  - λ is called a **variational parameter**

Variational Methods: Tangent

- For any λ and any x:

$$-\ln x \geq -\lambda x + \ln \lambda + 1$$

where the right-hand side is a family of functions $g_\lambda(x)$



- For any x, there is some λ for which the bound is tight
  - λ is called a **variational parameter**
- In our case $\ln Z(\theta)$ is like $\ln x$ and Q is like λ

Variational Methods: Optimizing the Energy Functional

$$\max_{Q} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- What is the space of distributions $\mathcal{Q}$ that we are optimizing over?
  - Define an "easy" family of distributions $\mathcal{Q}$
  - Assume a factorized form that offers convenient structure
- The objective function is concave in Q, **but** there are exponentially many distributions $Q(x)$

Variational Methods: Optimizing the Energy Functional

$$\max_{Q} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- What is the space of distributions $\mathcal{Q}$ that we are optimizing over?
  - Define an "easy" family of distributions $\mathcal{Q}$
  - Assume a factorized form that offers convenient structure
- The objective function is concave in Q, **but** there are exponentially many distributions $Q(x)$
- Two general approaches:
  1. Optimize the *exact* energy functional, but restricted to a space of (simpler) distributions (that generally do not include P)
  2. Optimize an *approximate* energy functional

Variational Methods: Optimizing the Energy Functional

$$\max_{Q} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- What is the space of distributions $\mathcal{Q}$ that we are optimizing over?
  - Define an "easy" family of distributions $\mathcal{Q}$
  - Assume a factorized form that offers convenient structure
- The objective function is concave in Q, **but** there are exponentially many distributions $Q(x)$
- **Relaxation** algorithms (today) operate directly on *pseudomarginals* that may not be consistent with any joint distribution (approximate energy functional)
- **Structured variational** algorithms (next lecture) optimize the exact energy functional over a family $\mathcal{Q}$ of tractable, *coherent* distributions

Exact Inference as Optimization

- Consider a cluster tree $\mathcal{T}$ for P
- Suppose that we have a set of beliefs over clusters and sepsets in $\mathcal{T}$

$$\mathcal{Q} = \{\beta_i : i \in V_{\mathcal{T}}\} \cup \{\mu_{i,j} : (i - j) \in E_{\mathcal{T}}\}$$

- Recall that the set of beliefs defines a distribution

$$Q = \frac{\prod_{i \in V_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{ij})}$$

- If Q is calibrated, the beliefs are the marginals

$$\beta_i(\boldsymbol{C}_i) = Q(\boldsymbol{S}_i)$$
$$\mu_{i,j}(\boldsymbol{S}_{i,j}) = Q(\boldsymbol{S}_{i,j})$$

Exact Inference as Optimization

- Given clique tree $\mathcal{T}$, search over the set of calibrated distributions $\mathcal{Q} = \{\beta_i : i \in V_\mathcal{T}\} \cup \{\mu_{i,j} : (i-j) \in E_\mathcal{T}\}$ defined over $\mathcal{T}$
- Suggests a constrained optimization:

$$\max_{Q \in \mathcal{Q}} F[\tilde{P}, Q]$$

subject to the constraints

$$\mu_{i,j}(\boldsymbol{s}_{i,j}) = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{c}_i) \quad \forall (i-j) \in E_\mathcal{T},\ \forall \boldsymbol{s}_{i,j} \in \mathsf{Val}(S_{i,j})$$

$$\sum_{\boldsymbol{c}_i} \beta_i(\boldsymbol{c}_i) = 1 \quad \forall i \in V_\mathcal{T}$$

- If $\mathcal{T}$ is an I-map for P, then there is *unique* solution $Q = P$ via sum-product (belief update) message passing

Exact Inference as Optimization

- Let's consider the *factored energy functional*

$$\tilde{F}[\tilde{P}, Q] = \sum_{i \in V_{\mathcal{T}}} \mathbb{E}_{\beta_i}[\ln \psi_i] + \sum_{i \in V_{\mathcal{T}}} H_{\beta_i}(\boldsymbol{C}_i) - \sum_{(i-j) \in E_{\mathcal{T}}} H_{\mu_{i,j}}(\boldsymbol{S}_{i,j})$$

- In general, $\tilde{F}[\tilde{P}, Q]$ is an approximation of the energy functional

Exact Inference as Optimization

- Let's consider the *factored energy functional*

$$\tilde{F}[\tilde{P}, Q] = \sum_{i \in V_{\mathcal{T}}} \mathbb{E}_{\beta_i}[\ln \psi_i] + \sum_{i \in V_{\mathcal{T}}} H_{\beta_i}(\boldsymbol{C}_i) - \sum_{(i-j) \in E_{\mathcal{T}}} H_{\mu_{i,j}}(\boldsymbol{S}_{i,j})$$

- In general, $\tilde{F}[\tilde{P}, Q]$ is an approximation of the energy functional
- If $\mathcal{Q}$ is a set of calibrated beliefs for $\mathcal{T}$ and

$$Q = \frac{\prod_{i \in V_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{ij})},$$

then the factored and original energy functionals are equivalent

$$\tilde{F}[\tilde{P}, Q] = F[\tilde{P}, Q]$$

Exact Inference as Optimization

- Let's consider the *factored energy functional*

$$\tilde{F}[\tilde{P}, Q] = \sum_{i \in V_{\mathcal{T}}} \mathbb{E}_{\beta_i}[\ln \psi_i] + \sum_{i \in V_{\mathcal{T}}} H_{\beta_i}(\boldsymbol{C}_i) - \sum_{(i-j) \in E_{\mathcal{T}}} H_{\mu_{i,j}}(\boldsymbol{S}_{i,j})$$

- In general, $\tilde{F}[\tilde{P}, Q]$ is an approximation of the energy functional
- If $\mathcal{Q}$ is a set of calibrated beliefs for $\mathcal{T}$ and

$$Q = \frac{\prod_{i \in V_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{ij})},$$

then the factored and original energy functionals are equivalent

$$\tilde{F}[\tilde{P}, Q] = F[\tilde{P}, Q]$$

Proof: $\sum_i \mathbb{E}_{\beta_i}[\ln \psi_i] = \sum_\phi \mathbb{E}_Q[\ln \phi]$ and

$$H_Q(\boldsymbol{X}) = \sum_{i \in V_{\mathcal{T}}} H_{\beta_i}(\boldsymbol{C}_i) - \sum_{(i-j) \in E_{\mathcal{T}}} H_{\mu_{i,j}}(\boldsymbol{S}_{i,j})$$

Exact Inference as Optimization

- Let's consider the *factored energy functional*

$$\tilde{F}[\tilde{P}, Q] = \sum_{i \in V_{\mathcal{T}}} \mathbb{E}_{\beta_i}[\ln \psi_i] + \sum_{i \in V_{\mathcal{T}}} H_{\beta_i}(\boldsymbol{C}_i) - \sum_{(i-j) \in E_{\mathcal{T}}} H_{\mu_{i,j}}(\boldsymbol{S}_{i,j})$$

- In general, $\tilde{F}[\tilde{P}, Q]$ is an approximation of the energy functional
- If $\mathcal{Q}$ is a set of calibrated beliefs for $\mathcal{T}$ and

$$Q = \frac{\prod_{i \in V_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{ij})},$$

  then the factored and original energy functionals are equivalent

$$\tilde{F}[\tilde{P}, Q] = F[\tilde{P}, Q]$$

- Whereas $F[\tilde{P}, Q]$ involves the entropy of the full joint distribution, $\tilde{F}[\tilde{P}, Q]$ involves (local) entropies of clusters and sepsets

Exact Inference as Optimization

- Gives rise to an alternative optimization

$$\max_{Q \in \mathcal{Q}} \tilde{F}[\tilde{P}, Q]$$

subject to the constraints

$$\mu_{i,j}(\boldsymbol{s}_{i,j}) = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{c}_i) \quad \forall (i-j) \in E_{\mathcal{T}},\ \forall \boldsymbol{s}_{i,j} \in \mathsf{Val}(S_{i,j})$$

$$\sum_{\boldsymbol{c}_i} \beta_i(\boldsymbol{c}_i) = 1 \quad \forall i \in V_{\mathcal{T}}$$

where $\mathcal{Q} = \{\beta_i : i \in V_{\mathcal{T}}\} \cup \{\mu_{i,j} : (i-j) \in E_{\mathcal{T}}\}$ is the space of calibrated sets

Exact Inference as Optimization

- **Theorem**: There is a stationary point iff there is a set of factors $\{m_{i \to j}(\boldsymbol{S}_{i,j}) : (i-j) \in E_{\mathcal{T}}\}$ such that

$$m_{i \to j}(\boldsymbol{S}_{i,j}) \propto \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \left(\prod_{k \in \mathsf{Nb}_i - \{j\}} m_{k \to i} \right)$$

from which we can calculate

$$\beta_i \propto \psi_i \left(\prod_{k \in \mathsf{Nb}_i - \{j\}} m_{k \to i} \right)$$

$$\mu_{i,j} \propto m_{j \to i} \cdot m_{j \to i}$$

- These fixed-point equations specify the relationship between parameters at the optimum
- They provide an iterative means of solving for the messages

Exact Inference as Optimization: Lagrangian

- We add two types of Lagrange multipliers, one for each constraint

$$L = \tilde{F}[\tilde{P}, Q] - \sum_{i \in V_{\mathcal{T}}} \lambda_i \left(\sum_{\boldsymbol{c}_i} \beta_i(\boldsymbol{c}_i) - 1 \right)$$
$$- \sum_i \sum_{j \in \mathsf{Nb}_i} \sum_{\boldsymbol{s}_{i,j}} \lambda_{j \to i}(\boldsymbol{s}_{i,j}) \left(\sum_{\boldsymbol{c}_i \sim \boldsymbol{s}_{i,j}} \beta_i(\boldsymbol{c}_i) - \mu_{i,j}(\boldsymbol{s}_{i,j}) \right)$$

- The Lagrangian is a function of $\{\beta_i\}$, $\{\mu_{i,j}\}$, and the Lagrange multipliers $\{\lambda_i\}$ and $\{\lambda_{i \to j}\}$
- To find the maximum of the Lagrangian, we take partial derivatives w.r.t. $\{\beta_i\}$, $\{\mu_{i,j}\}$, and the Lagrange multipliers

Exact Inference as Optimization: Stationary Points

- The partial derivatives become

$$\frac{\partial L}{\partial \beta_i(\boldsymbol{c}_i)} = \ln \psi_i(\boldsymbol{c}_i) - \ln \beta_i(\boldsymbol{c}_i) - 1 - \lambda_i - \sum_{j \in \mathsf{Nb}_i} \lambda_{j \to i}(\boldsymbol{s}_{i,j})$$

$$\frac{\partial L}{\partial \mu_{i,j}(\boldsymbol{s}_{i,j})} = \ln \mu_{i,j}(\boldsymbol{s}_{i,j}) + 1 + \lambda_{i \to j}(\boldsymbol{s}_{i,j}) + \lambda_{j \to i}(\boldsymbol{s}_{i,j})$$

- At the stationary point, these derivatives are zero, yielding

$$\beta_i(\boldsymbol{c}_i) = \exp(1 - \lambda_i)\psi_i(\boldsymbol{c}_i) \prod_{j \in \mathsf{Nb}_i} \exp(-\lambda_{j \to i}(\boldsymbol{s}_{i,j}))$$

$$\mu_{i,j}(\boldsymbol{s}_{i,j}) = \exp(-1) \exp(-\lambda_{i \to j}(\boldsymbol{s}_{i,j})) \exp(-\lambda_{j \to i}(\boldsymbol{s}_{i,j}))$$

- We can convert these to messages by defining

$$m_{i \to j}(\boldsymbol{s}_{i,j}) = \exp\left(-\lambda_{i \to j}(\boldsymbol{s}_{i,j}) - \frac{1}{2}\right)$$

Exact Inference as Optimization: Stationary Points

- The equations become

$$\beta_i(\boldsymbol{c}_i) = \exp\left(-\lambda_i - 1 + \frac{1}{2}|\mathsf{Nb}_i|\right)\psi_i(\boldsymbol{c}_i)\prod_{j\in\mathsf{Nb}_i} m_{j\to i}(\boldsymbol{s}_{i,j})$$

$$\mu_{i,j}(\boldsymbol{s}_{i,j}) = m_{i\to j}(\boldsymbol{s}_{i,j})\cdot m_{j\to i}(\boldsymbol{s}_{i,j})$$

- The messages are given by

$$m_{i\to j}(\boldsymbol{s}_{i,j}) = \frac{\mu_{i,j}(\boldsymbol{s}_{i,j})}{m_{j\to i}(\boldsymbol{s}_{i,j})} = \frac{\sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}}\beta_i(\boldsymbol{C}_i, \boldsymbol{s}_{i,j})}{m_{j\to i}(\boldsymbol{s}_{i,j})}$$

$$= \exp\left(-\lambda_i - 1 + \frac{1}{2}|\mathsf{Nb}_i|\right)\sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}}\psi_i(\boldsymbol{c}_i)\prod_{k\in\mathsf{Nb}_i - \{j\}} m_{k\to i}(\boldsymbol{s}_{i,k})$$

where the first term is a normalizing constant

Exact Inference as Optimization: Stationary Points

Theorem

A set of beliefs $\boldsymbol{Q}$ is a stationary point iff there exists $\{m_{i\to j}(\boldsymbol{s}_{i,j}) : (i-j) \in E_{\mathcal{T}}\}$ *such that*

$$m_{i\to j}(\boldsymbol{s}_{i,j}) \propto \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i(\boldsymbol{c}_i) \prod_{k \in Nb_i - \{j\}} m_{k\to i}(\boldsymbol{s}_{i,k})$$

with

$$\beta_i \propto \psi_i \left(\prod_{j \in Nb_i} m_{i\to j} \right)$$

$$\mu_{i,j} = m_{j\to i} \cdot m_{i\to j}$$

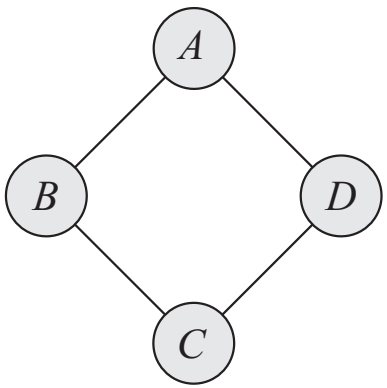
Inference as Optimization

$$m_{i \to j}(\boldsymbol{S}_{i,j}) \propto \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \left(\prod_{k \in \mathsf{Nb}_i - \{j\}} m_{k \to i} \right)$$

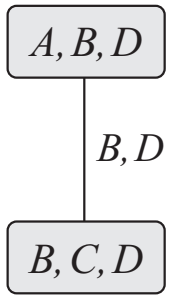
- Fixed-point characterization suggests iterative optimization process
  1. Initialize $m_{i \to j} = 1$ for all $(i - j)$
  2. Update each $m_{i \to j}$ to equal RHS of above equation
  3. Repeat *towards* convergence
- This looks awfully like message passing. . .
- Will converge under certain conditions (e.g., over a clique tree)

Loopy Cluster Graphs

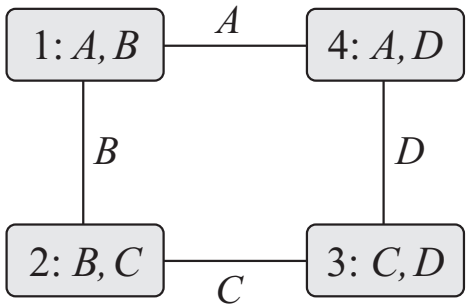
- Clique trees are a useful data structure for exact inference
  - Groups of random variables (cliques) as nodes
  - Edge structure forms a tree
  - Exhibits running intersection property (sepsets are intersections)
- Cluster graphs
  - Can have loops (need not be a tree)
  - The clusters tend to be smaller
  - Exhibits a modified version of the running intersection property



MRF



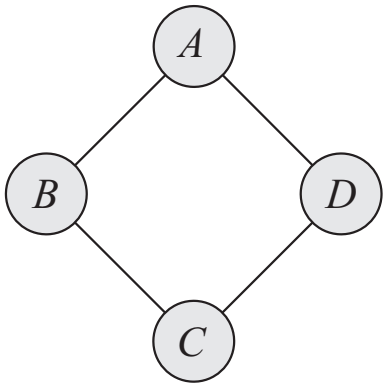
Clique Tree



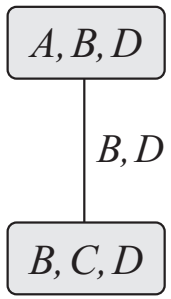
Cluster Graph

Propagation-Based Approximation

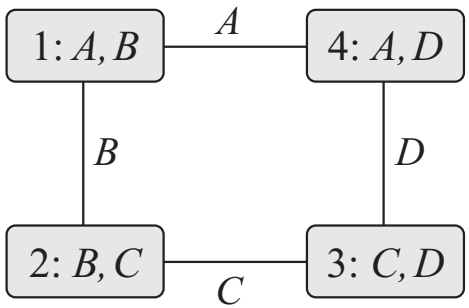
- Belief update message passing algorithm does not rely on the graph being a tree (neither does sum-product w/ some tricks)
- We can consider using the sum-product and belief-update message passing algorithms on cluster graphs (even with cycles)
- When there are cycles, this is known as **loopy belief propagation**



MRF



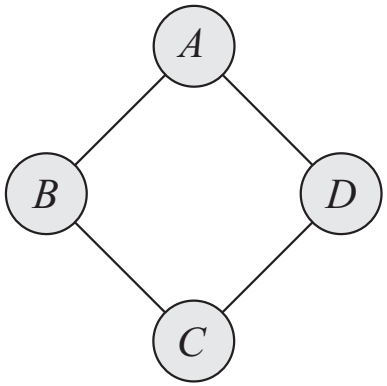
Clique Tree



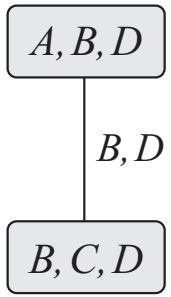
Cluster Graph

Propagation-Based Approximation: Effects

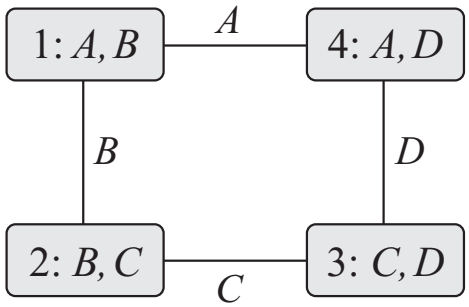
- Smaller clusters mean faster message computation
- Guarantee of two-pass convergence is gone (unless it is a clique tree)
- In fact, it isn't clear that we have *any* convergence guarantees
- Message passing with loops results in double-counting information
- Resulting node beliefs may not equate to marginals



MRF



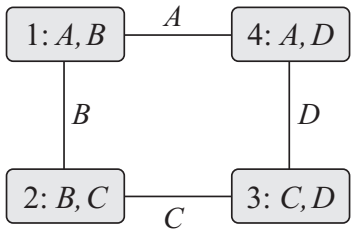
Clique Tree



Cluster Graph

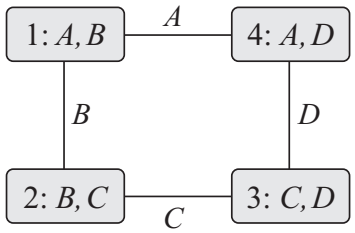
Propagation-Based Approximation: Example

- Consider the previous MRF for binary-valued variables with potentials that encourage joint assignments (i.e., $A = B$)
- Suppose we perform message passing over the cluster graph



Propagation-Based Approximation: Example

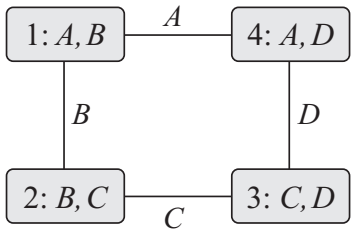
- Consider the previous MRF for binary-valued variables with potentials that encourage joint assignments (i.e., $A = B$)
- Suppose we perform message passing over the cluster graph



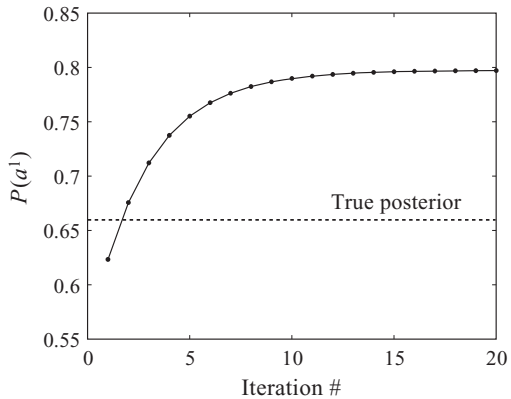
- If $m_{1\to 2}$ encourages $B = 1$, then $m_{2\to 3}$ encourages $C = 1$, $\ldots$ $m_{4\to 1}$ encourages $A = 1$, thereby double-counting evidence

Propagation-Based Approximation: Example

- Consider the previous MRF for binary-valued variables with potentials that encourage joint assignments (i.e., $A = B$)
- Suppose we perform message passing over the cluster graph

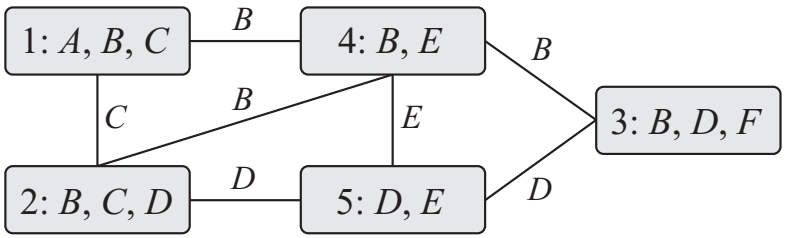


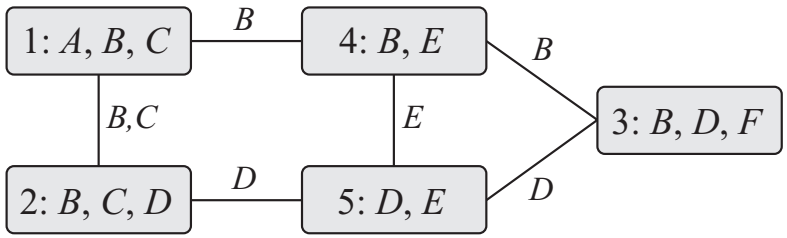
- If $m_{1\to 2}$ encourages $B = 1$, then $m_{2\to 3}$ encourages $C = 1, \ldots m_{4\to 1}$ encourages $A = 1$, thereby double-counting evidence
- Resulting belief over-estimates marginal over A



Modified Running Intersection Property

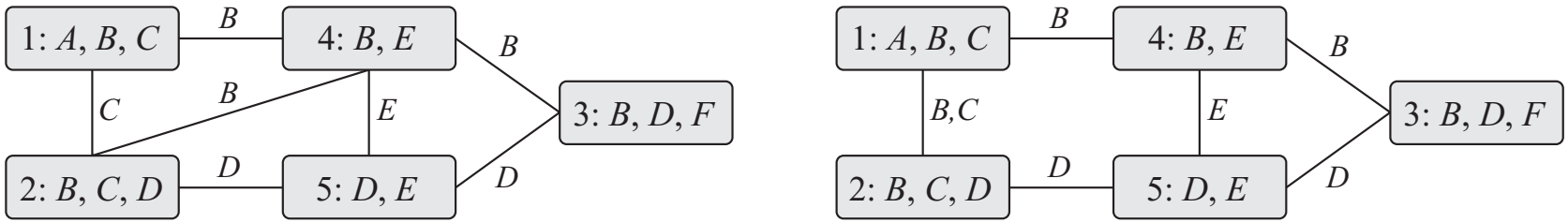
- For every X for which $X_i \in \boldsymbol{C}_i$ and $X_i \in \boldsymbol{C}_j$, there is a *single* path between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ for which $X \in \boldsymbol{S}_e$ along all edges in path
  - There may be other paths that connect the nodes
  - Unlike with clique trees, this does *not* imply that $\boldsymbol{S}_{i,j} = \boldsymbol{C}_i \cap \boldsymbol{C}_j$
  - Instead $\boldsymbol{S}_{i,j} \subseteq \boldsymbol{C}_i \cap \boldsymbol{C}_j$





Modified Running Intersection Property

- For every X for which $X_i \in \boldsymbol{C}_i$ and $X_i \in \boldsymbol{C}_j$, there is a *single* path between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ for which $X \in \boldsymbol{S}_e$ along all edges in path
  - There may be other paths that connect the nodes
  - Unlike with clique trees, this does *not* imply that $\boldsymbol{S}_{i,j} = \boldsymbol{C}_i \cap \boldsymbol{C}_j$
  - Instead $\boldsymbol{S}_{i,j} \subseteq \boldsymbol{C}_i \cap \boldsymbol{C}_j$



- We say that a cluster graph is *calibrated* if adjacent clusters agree on the belief over their sepset (not intersection)

$$\sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i = \sum_{\boldsymbol{C}_j - \boldsymbol{S}_{i,j}} \beta_j$$

The marginal for a variable X is identical in all clusters containing X

Cluster Graph (Loopy) Belief Propagation

- Can apply both sum-product and belief update message passing
- But . . . sum-product requires that nodes receive messages (i.e., be *ready*) before transmitting
  - Initialize all messages to $m_{i \to j} = 1$
- Keep sending messages until graph is calibrated
- Can be significantly cheaper than exact inference (e.g., quadratic vs. exponential in n for an $n \times n$ pairwise MRF)

Algorithm 22.1: Loopy belief propagation for a pairwise MRF

Input: node potentials $\psi_s(x_s)$, edge potentials $\psi_{st}(x_s, x_t)$;
Initialize messages $m_{s \to t}(x_t) = 1$ for all edges $s - t$;
Initialize beliefs $\text{bel}_s(x_s) = 1$ for all nodes s;
repeat
  Send message on each edge
  $m_{s \to t}(x_t) = \sum_{x_s} \left(\psi_s(x_s) \psi_{st}(x_s, x_t) \prod_{u \in \text{nbr}_s \setminus t} m_{u \to s}(x_s) \right)$;
  Update belief of each node $\text{bel}_s(x_s) \propto \psi_s(x_s) \prod_{t \in \text{nbr}_s} m_{t \to s}(x_s)$;
until *beliefs don't change significantly*;
Return marginal beliefs $\text{bel}_s(x_s)$;

Cluster Graph Invariant

- As with sum-product (and belief update) over clique trees, LBP is invariant throughout the algorithm: For cluster graph $\mathcal{U}$ with beliefs $\{\beta_i\}$ and sepsets $\{\mu_{i,j}\}$

$$\tilde{P}(\mathcal{X}) = \frac{\prod_{i \in V_{\mathcal{U}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{U}}} \mu_{i,j}(\boldsymbol{S}_{i,j})}$$

- However, beliefs are not necessarily marginals of P, i.e., $\beta_i(\boldsymbol{C}_i) \neq P(\boldsymbol{C}_i)$

Cluster Graph Invariant

- As with sum-product (and belief update) over clique trees, LBP is invariant throughout the algorithm: For cluster graph $\mathcal{U}$ with beliefs $\{\beta_i\}$ and sepsets $\{\mu_{i,j}\}$

$$\tilde{P}(\mathcal{X}) = \frac{\prod_{i \in V_{\mathcal{U}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{U}}} \mu_{i,j}(\boldsymbol{S}_{i,j})}$$

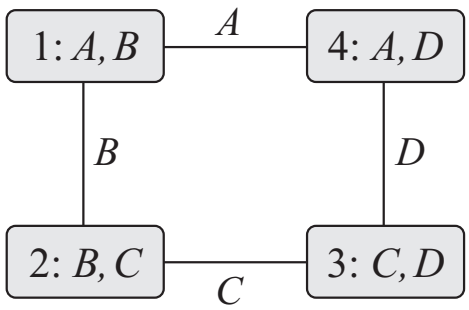
- However, beliefs are not necessarily marginals of P, i.e., $\beta_i(\boldsymbol{C}_i) \neq P(\boldsymbol{C}_i)$
- Instead, we can consider *any* subtree $\mathcal{T}$ of $\mathcal{U}$ that satisfies running intersection property, thereby defining the distribution

$$P_{\mathcal{T}}(\mathcal{X}) = \frac{\prod_{i \in V_{\mathcal{T}}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in E_{\mathcal{T}}} \mu_{i,j}(\boldsymbol{S}_{i,j})}$$

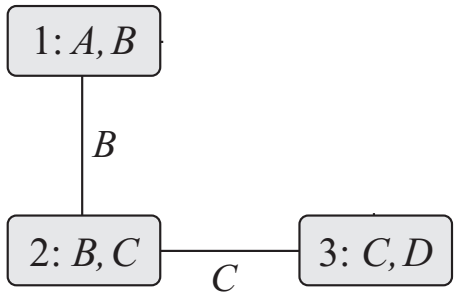
- The beliefs over $\boldsymbol{C}_i$ are marginals of $P_{\mathcal{T}}$, i.e.,

$$\beta_i(\boldsymbol{C}_i) = P_{\mathcal{T}}(\boldsymbol{C}_i)$$

Cluster Graph Invariant: Example



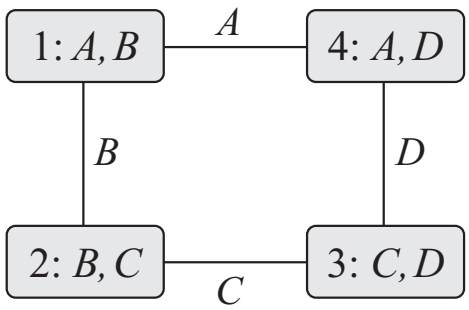
Cluster Graph



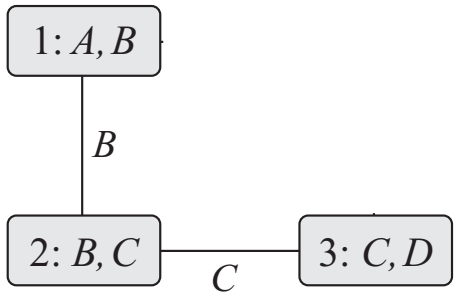
Subtree

- Let's revisit the pairwise MRF over four random variables
- The tree on the right is a valid subtree (running intersection property)

Cluster Graph Invariant: Example



Cluster Graph

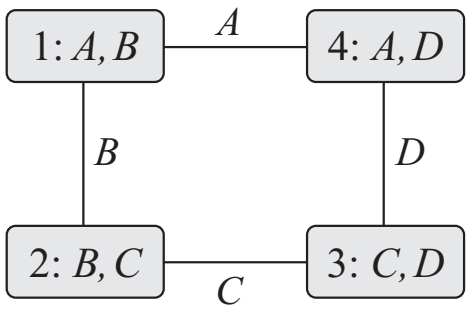


Subtree

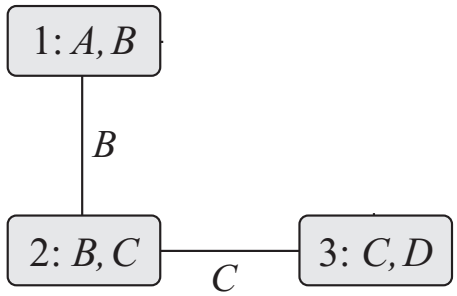
- Let's revisit the pairwise MRF over four random variables
- The tree on the right is a valid subtree (running intersection property)
- When the tree is calibrated, $\beta_1(A, B) = P_{\mathcal{T}}(A, B)$ and we can show

$$P_{\mathcal{T}}(A, B, C, D) = P_{\Phi}(A, B, C, D) \frac{\mu_{3,4}(D)\mu_{1,4}(A)}{\beta_4(A, D)}$$

Cluster Graph Invariant: Example



Cluster Graph



Subtree

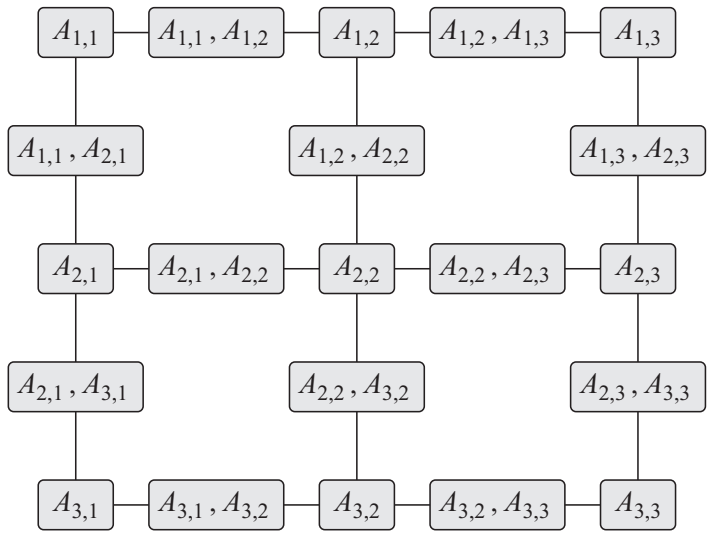
- Let's revisit the pairwise MRF over four random variables
- The tree on the right is a valid subtree (running intersection property)
- When the tree is calibrated, $\beta_1(A, B) = P_{\mathcal{T}}(A, B)$ and we can show

$$P_{\mathcal{T}}(A, B, C, D) = P_{\Phi}(A, B, C, D)\frac{\mu_{3,4}(D)\mu_{1,4}(A)}{\beta_4(A, D)}$$

- $\frac{\mu_{3,4}(D)\mu_{1,4}(A)}{\beta_4(A,D)}$ is the **cluster graph residual** that measures the error in the marginal distribution relative to P_{Φ}

Constructing Cluster Graphs

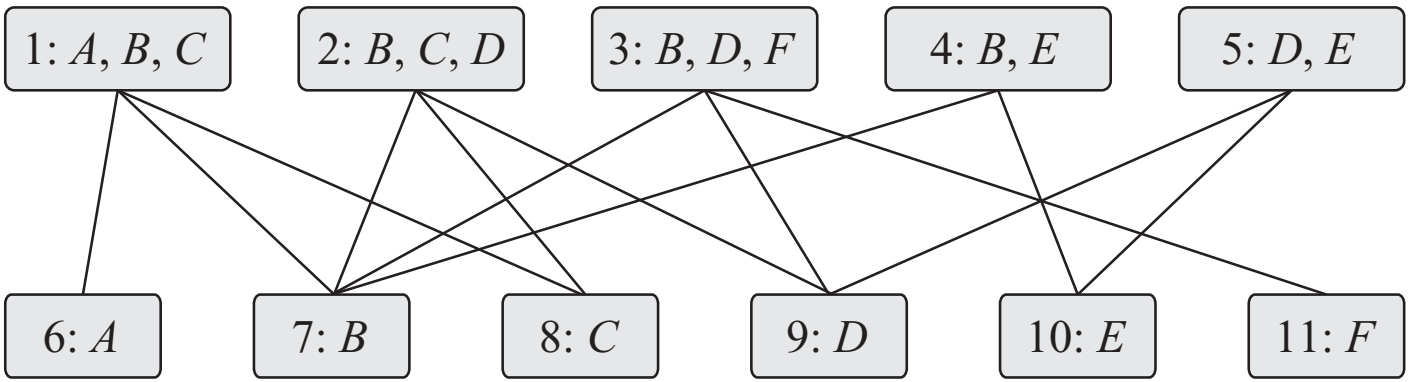
- Different graphs result in different efficiency and acuracy
- Consider a pairwise Markov network, i.e., one having unary $\phi_i(X_i)$ and pairwise $\phi_{i,j}(X_i, X_j)$ factors
- One possible cluster graph follows from introducing a cluster for each potential and adding edges between clusters with overlapping scope



- $\boldsymbol{C}_{i,j}$ acts as a conduit for messages between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$, so we can think of messages passing as operating on original MRF (as in PSet 2)

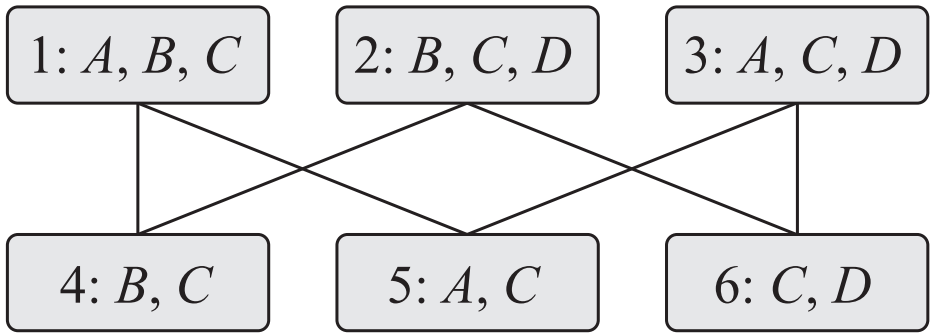
Constructing Cluster Graphs: Bethe Cluster Graphs

- A generalization of the above approach to more complex graphs
- A bipartite graph where
  - The first layer consists of one cluster for each factor $\phi \in \Phi$
  - Second layer consists of one cluster for each variable
  - Add edges between each univariate cluster $\phi_i(X)$ and each cluster in the first layer with X in its scope
- Satisfies the modified running intersection property
- Drawback: Messages are passed through univariate factors, losing information about interactions



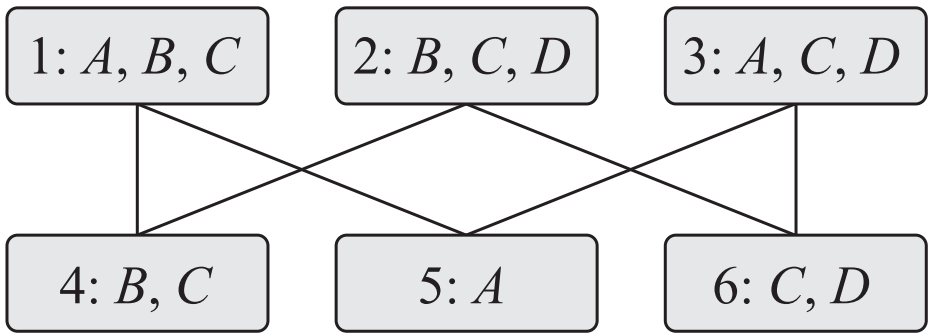
Constructing Cluster Graphs

- One possibility would be to preserve pairwise interactions by replacing lower-layer with pairwise clusters
- However, this graph doesn't satisfy the running intersection property (each edge involves C)



Constructing Cluster Graphs

- One possibility would be to preserve pairwise interactions by replacing lower-layer with pairwise clusters
- However, this graph doesn't satisfy the running intersection property (each edge involves C)
- We can remedy this via a weaker approximation that removes C from one cluster



Cluster Graph Belief Propagation

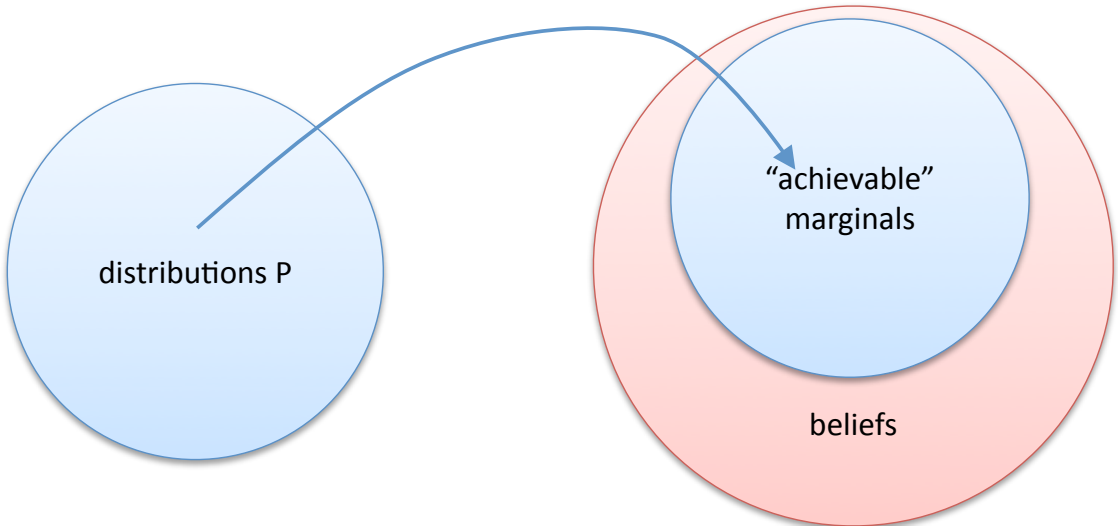
- Energy functional has many local optima
- Cluster graph belief propagation does not necessarily converge
- May exhibit oscillations
- Problems tend to be worse for "peakier" or more deterministic models
- "Dampening" the messages can help to avoid oscillations

$$m_{i \to j}(\boldsymbol{S}_{i,j}) \leftarrow \lambda \left(\sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \prod_{k \neq i} m_{k \to i} \right) + (1 - \lambda) m_{i \to j}^{\text{old}} \quad \lambda \in (0, 1]$$

- In practice, asynchronous message passing works better than synchronous message passing
- A number of other techniques exist to improve convergence

Variational Analysis

- We defined the factored energy functional $\tilde{F}[\tilde{P}, Q]$ in terms of entropies of clusters and sepsets, which makes it easy to optimize
- However, unlike with clique trees, it is now an *approximation* to the energy functional
- Consequently, it is no longer a bound on the (log) partition function (which we have been maximizing to better approximate P_Φ)
- Further, it is possible to have a calibrated cluster graph whose beliefs are not *globally* consistent, i.e., they don't represent the marginal of any single joint distribution over $\mathcal{X}$



Variational Analysis

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Let $\boldsymbol{\mu}_Q$ be the vector of *marginals* of $Q(\boldsymbol{X})$
- We can express the objective as

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{x}_c} \theta_c(\boldsymbol{X}_c) \mu_Q^c(\boldsymbol{x}_c) + H(\boldsymbol{\mu}_Q)$$

  where $H(\boldsymbol{\mu}_Q)$ is the entropy of the *maximum entropy distribution* with marginals $\boldsymbol{\mu}_Q$
- Rather than optimize over Q, optimize over *valid* marginal vectors $\boldsymbol{\mu}_Q$
- But, what is the space that we are optimizing over?

Variational Analysis

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Let $\boldsymbol{\mu}_Q$ be the vector of *marginals* of $Q(\boldsymbol{X})$
- We can express the objective as

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{x}_c} \theta_c(\boldsymbol{X}_c) \mu_Q^c(\boldsymbol{x}_c) + H(\boldsymbol{\mu}_Q)$$

  where $H(\boldsymbol{\mu}_Q)$ is the entropy of the *maximum entropy distribution* with marginals $\boldsymbol{\mu}_Q$
- Rather than optimize over Q, optimize over *valid* marginal vectors $\boldsymbol{\mu}_Q$
- But, what is the space that we are optimizing over?

They must be valid marginals of a single distributions

Marginal Polytope

$$\max_{\boldsymbol{\mu}_Q \in \mathcal{M}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{x}_c} \theta_c(\boldsymbol{X}_c) \mu_Q^c(\boldsymbol{x}_c) + H(\boldsymbol{\mu}_Q)$$

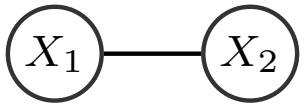
- The **marginal polytope** is the set of marginals (cluster and sepset beliefs) associated with an actual distribution P

$$\mathcal{M} = \left\{ \boldsymbol{\mu}_Q \,|\, \exists\, P \text{ s.t. } \boldsymbol{\mu}_Q^c = \sum_{\boldsymbol{X} \setminus \boldsymbol{X}_c} P(\boldsymbol{X}) \;\forall\; \boldsymbol{\mu}_Q^c \right\}$$

- It is the set of *marginals* obtainable from the *polytope* of distributions over $\mathcal{X}$
- The marginal polytope is a convex set

(We should really be presenting this in terms of exponential family)

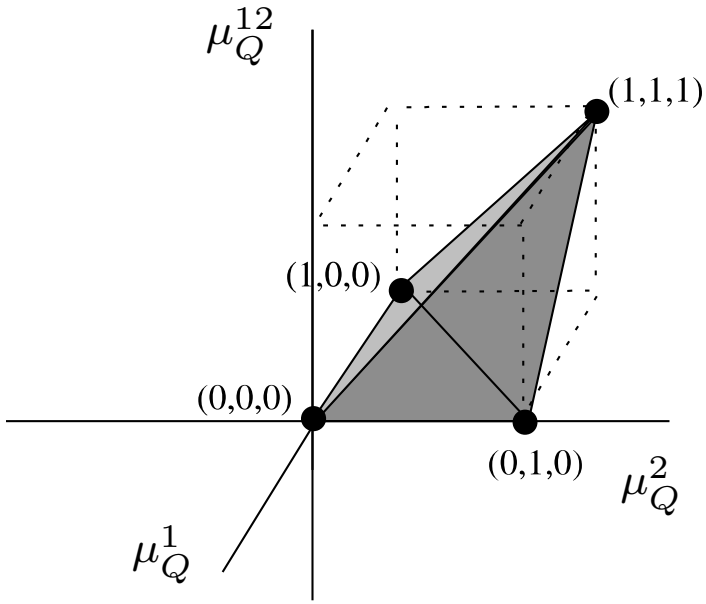
Marginal Polytope: Example



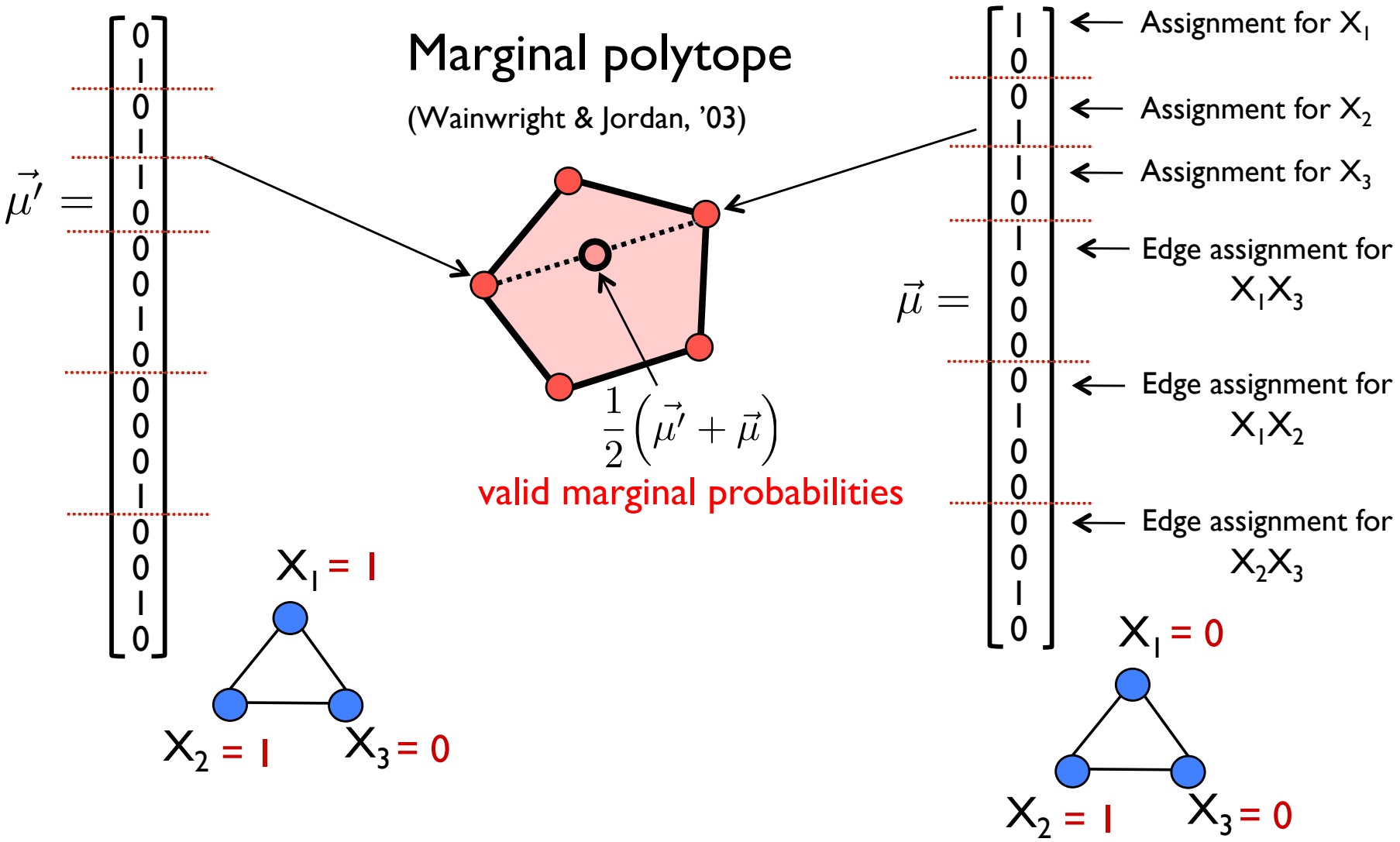
- Consider a two-node Ising model (i.e., $X_i \in \{0, 1\}$)
- The relevant marginals are

$$\mu_Q^1(X_1) = P(X_1 = 1), \mu_Q^2(X_2) = P(X_2 = 1),$$
$$\text{and } \mu_Q^{1,2}(X_1, X_2) = P(X_1 = 1, X_2 = 1)$$

- $\mathcal{M} = \text{conv}\left[(0,0,0), (1,0,0), (0,1,0), (1,1,1)\right]$



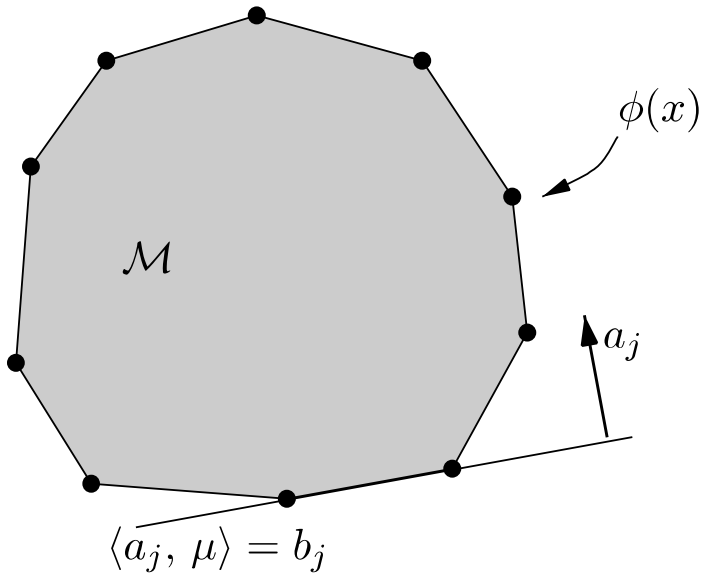
Variational Analysis: Marginal Polytope



Marginal Polytope

- The marginal polytope is convex
- Half-plane representation: By the **Minkowski-Weyl Theorem**, any non-empty polytope can be defined by a *finite* collection of linear equality constraints

$$\mathcal{M} = \left\{ \boldsymbol{\mu} \in \mathbb{R}^d \,|\, \boldsymbol{a}_j^\top \boldsymbol{\mu} \geq b_j, \forall j \in \mathcal{J} \right\}$$



Relaxation

$$\max_{\boldsymbol{\mu}_Q \in \mathcal{M}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{x}_c} \theta_c(\boldsymbol{X}_c) \mu_Q^c(\boldsymbol{x}_c) + H(\boldsymbol{\mu}_Q)$$

- However, we haven't made progress because
  1. The marginal polytope $\mathcal{M}$ not compact (in general, there are exponentially many vertices and facets)
  2. $H(\boldsymbol{\mu}_Q)$ is very difficult to compute and optimize over
  3. Optimizing over the polytope is as hard as inference
- Thus, we make two approximations:
  1. We replace $\mathcal{M}$ with a relaxation of the marginal polytope
  2. We replace $H(\boldsymbol{\mu}_Q)$ with an approximation $\tilde{H}(\boldsymbol{\mu}_Q)$

Relaxation: Optimization over Local Consistency Polytope

- Instead, perform inference over the **local consistency polytope** $\text{Local}[\mathcal{U}] =$

$$\left\{ \begin{array}{c} \{\beta_i : i \in \mathcal{V}_\mathcal{U}\} \cup \\ \{\mu_{i,j} : (i\text{-}j) \in \mathcal{E}_\mathcal{U}\} \end{array} \middle| \begin{array}{lcll} \mu_{i,j}[\boldsymbol{s}_{i,j}] & = & \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{c}_i) & \forall (i\text{-}j) \in \mathcal{E}_\mathcal{U}, \forall \boldsymbol{s}_{i,j} \in Val(\boldsymbol{S}_{i,j}) \\ 1 & = & \sum_{\boldsymbol{c}_i} \beta_i(\boldsymbol{c}_i) & \forall i \in \mathcal{V}_\mathcal{U} \\ \beta_i(\boldsymbol{c}_i) & \geq & 0 & \forall i \in \mathcal{V}_\mathcal{U}, \boldsymbol{c}_i \in Val(\boldsymbol{C}_i). \end{array} \right\}$$

- Defines a set of *pseudomarginals*, each over variables in one cluster
- These pseudomarginals are locally consistent (calibrated), but they aren't necessarily the marginals of a single underlying joint distribution
- Results in an alternative optimization

$$\max_{Q \in \text{Local}[\mathcal{U}]} \tilde{F}[\tilde{P}, Q]$$

- This optimization employs two approximations:
  1. Approximate energy functional (no longer lower-bound on log-partition)
  2. Optimizing over the space of pseudo-marginals

Equivalence

- Convergence point of cluster graph belief propagation equates to a stationary point of the factored energy functional over the local consistency polytope