Probabilistic Graphical Models

Lecture 15 – Inference as Optimization

CS/CNS/EE 155

Andreas Krause

Announcements

- Homework 3 due next Monday (Nov 23)

- Project poster session on Friday December 4 (tentative)
- Final writeup (8 pages NIPS format) due Dec 9

Approximate inference

- Three major classes of general-purpose approaches

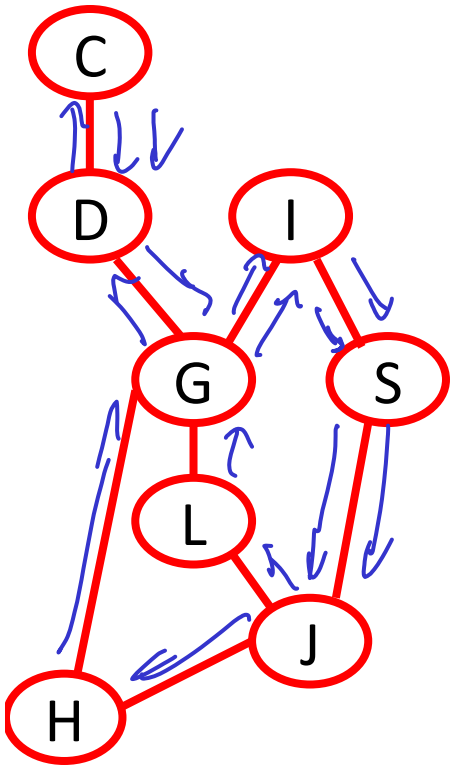
- **Message passing**
  - E.g.: Loopy Belief Propagation

- **Inference as optimization**
  - Approximate posterior distribution by simple distribution
  - Mean field / structured mean field

- **Sampling based inference**
  - Importance sampling, particle filtering
  - Gibbs sampling, MCMC

- Many other alternatives (often for special cases)

Loopy BP on arbitrary pairwise MNs



- What if we apply BP to a graph with loops?
  - Apply BP and hope for the best..

$$\delta_{i\to j}(X_j) = \sum_{x_i} \pi_i(x_i)\pi_{i,j}(x_i, X_j) \prod_{s\in N(i)\setminus\{j\}} \delta_{s\to i}(x_i)$$

- Will not generally converge.. ☹
- If it converges, will not necessarily get correct marginals ☹

- However, in practice, answers often still useful!

Approximate inference

- Three major classes of general-purpose approaches
- **Message passing**
  - E.g.: Loopy Belief Propagation (today!)
- **Inference as optimization**
  - Approximate posterior distribution by simple distribution
  - Mean field / structured mean field
  - Assumed density filtering / expectation propagation
- **Sampling based inference**
  - Importance sampling, particle filtering
  - Gibbs sampling, MCMC
- Many other alternatives (often for special cases)

Variational approximation

- Graphical model with intractable (high-treewidth) joint distribution $P(X_1,\ldots,X_n)$
- Want to compute posterior distributions

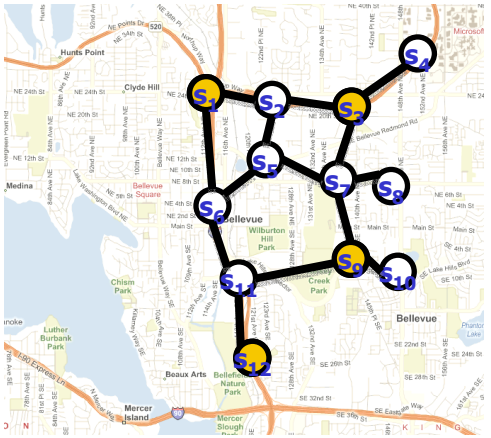
$$P(X_3, X_4 \mid X_1 = x_1, X_2 = x_2)$$

$$\propto P(X_3, X_4, X_1 = x_1, X_2 = x_2) = \sum_{x_1 x_2 x_5 \ldots} \sum \sum P(X_1, X_2, \ldots, X_N)$$

- Computing posterior exactly is intractable
- **Key idea**: Approximate posterior with *simpler* distribution that's *as close* to P as possible

Why should we hope that we can find a simple approximation?

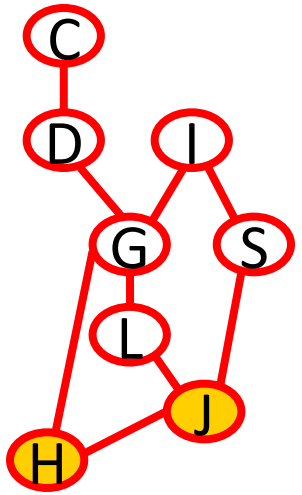
- Prior distribution is complicated
  - Need to describe all possible states of the world (and relationships between variables)



- Posterior distribution is often simple:
  - Have made many observations ➔ less uncertainty
  - Variables can become "almost independent"

- For now: Represent posterior as undirected model (and instantiate observations)

$$P(X_1, \ldots, X_n \mid obs) = \frac{1}{Z} \prod_j \Psi_j(\mathbf{C}_j)$$



Variational approximation

- **Key idea**: Approximate posterior with simpler distribution that's as close as possible to P
  - What is a "simple" distribution?
  - What does "as close as possible" mean?

- **Simple** = efficient inference
  - Typically: factorized (fully independent, chain, tree, …)
  - Gaussian approximation
- **As close as possible** = KL divergence (typically)
  - Other distance measures can be used too, but more challenging to compute

Kullback-Leibler (KL) divergence

- Distance between distributions

$$D(P||Q) = \int P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} d\mathbf{x}$$

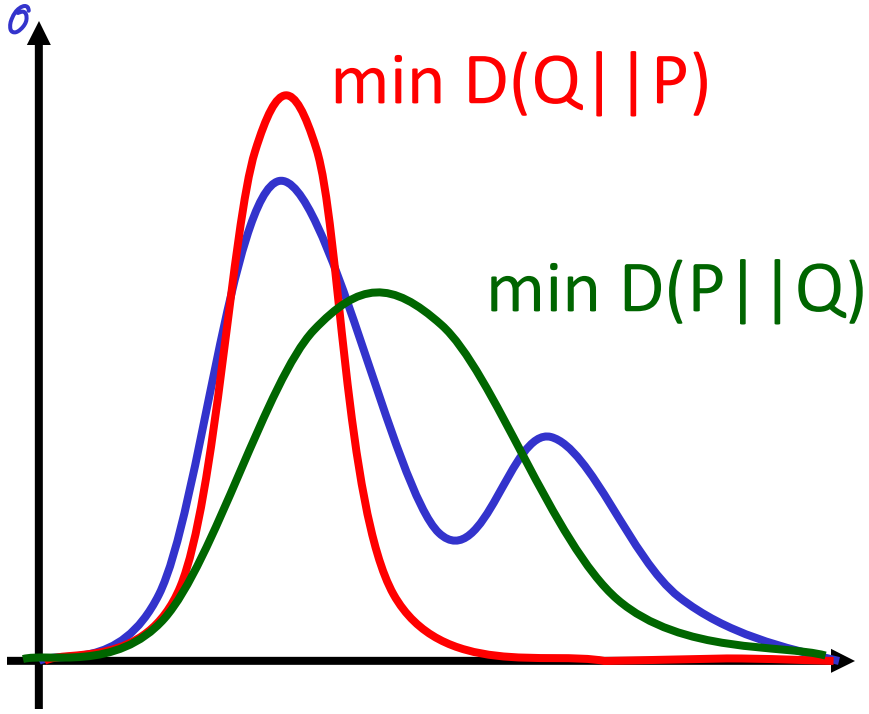
- Properties:
  - $D(P \,||\, Q) \geq 0$
  - P(x)=Q(x) almost everywhere $\Leftrightarrow$ $D(P \,||\, Q) = 0$
- In general, $D(P \,||\, Q) \neq D(Q \,||\, P)$
  - P determines when difference is important

$P(x) = 0$, $Q(x) \neq 0$ $\Rightarrow$ $0 \cdot \log \frac{Q}{\varepsilon} = 0$

$P(x) = \varepsilon$, $Q(x) = 0$ $\Rightarrow$ $\varepsilon \cdot \log \frac{\varepsilon}{0} = \infty$

Finding simple approximate distributions

- KL divergence not symmetric; need to choose directions
- P: true distribution; Q: our approximation
- D(P || Q)
  - The "right" way
  - Q chosen to "support" P
  - Often intractable to compute
- D(Q || P)
  - The "reverse" way
  - Underestimates support (overconfident)
  - Will be tractable to compute
- Both special cases of α-divergence



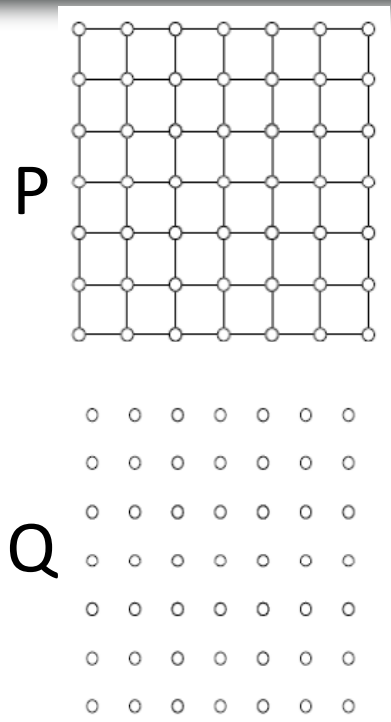
"Simple" distributions

- Simplest distribution: Q fully factorized
  - $Q(X_1,\ldots,X_n) = \prod_i Q_i(X_i)$

- M = {Q: Q fully factorized}
  = {Q: $Q(X) = \prod_i Q_i(X_i)$}

$$Q^* = \underset{Q \in \mathcal{M}}{\operatorname{argmin}} D(Q||P)$$

$D(P \| Q)$



- Can also find more structured approximations
  - Chains: $Q(X_1,\ldots,X_n) = \prod_i Q_i(X_i \mid X_{i-1})$
  - Trees
  - Any distributions one can do efficient inference on

Mean field approximation the "right way"

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

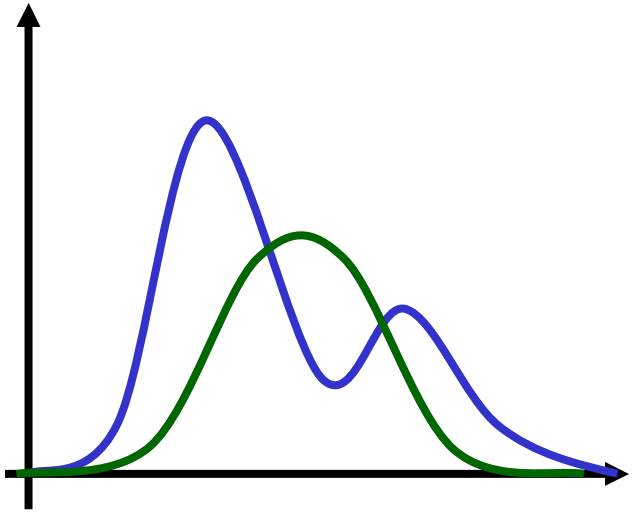
$$= \underbrace{\sum_x P(x) \log P(x)}_{\text{constant}} - \underbrace{\sum_x P(x) \log Q(x)}_{(*)}$$

$$(*) = \sum_x P(x) \log \prod_i Q_i(x_i)$$

$$= \sum_x P(x) \sum_i \log Q_i(x_i)$$

$$= \sum_i \sum_x P(x) \log Q_i(x_i)$$

$$= \sum_i \underbrace{\left(\sum_{x_i'} P(x_i') \log Q_i(x_i') \right)}_{\text{Need } P(x_i) \text{ intractable!}} \underbrace{\left(\sum_{x_{1:i-1,i+1:m}} P(x) \right)}_{1}$$



Mean field approximation the reverse way

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

$$= \underbrace{\sum_x Q(x) \log Q(x)}_{(1)} - \underbrace{\sum_x Q(x) \log P(x)}_{(2)}$$

$$(1) = -H(Q) = \sum_x Q(x) \log \prod_i Q_i(x_i)$$

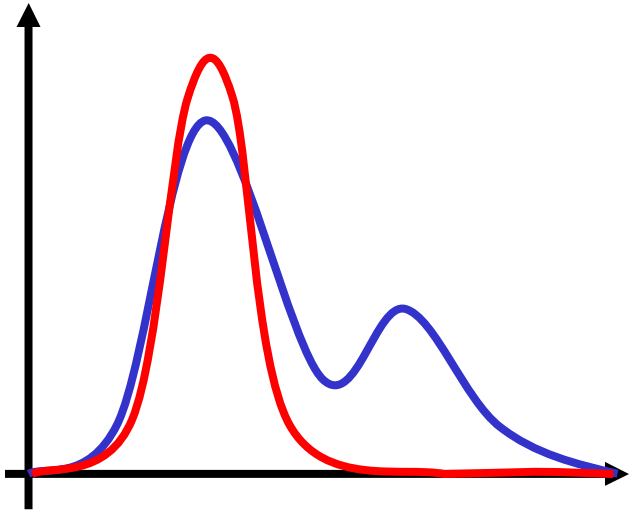
$$= \sum_i \sum_x Q(x) \log Q_i(x_i)$$

$$= \sum_i \sum_{x_i} Q_i(x_i) \log Q_i(x_i) \underbrace{\left(\sum_{x_{\neg i}} Q(x_{\neg i}) \right)}_{=1} = -\sum_i H(Q_i)$$

$$(2) = \sum_x Q(x) \log \frac{1}{Z} \prod_i \psi_i(c_i) = \underbrace{-\sum_x Q(x) \log Z}_{-\log Z} + \underbrace{\sum_x Q(x) \sum_i \log \psi_i(c_i)}_{(*)}$$

$$(*) = \sum_i \sum_x Q(x) \log \psi_i(c_i) = \sum_i \sum_{c_i} \underbrace{Q(c_i)}_{\text{Need marginal for } c_i} \log \psi_i(c_i)$$

$\psi_i(c_i)$: depends on small # of vars



Reverse KL for fully factorized case

$$D(Q||P) = -\sum_i \sum_x Q(x) \log \Psi_i(x) - \sum_i H(Q_i) + \ln Z$$

$\ln Z$: constant

$\sum_x Q(x) \log \Psi_i(x) = \mathbb{E}_Q[\log \Psi_i]$

$$\ln Z = D(Q||P) + \sum_i H(Q_i) + \sum_i \mathbb{E}_Q[\log \Psi_i]$$

$\ln Z$: constant

$D(Q||P)$: $\to$ min

$\sum_i H(Q_i) + \sum_i \mathbb{E}_Q[\log \Psi_i]$: $\to$ max

$F(Q; P)$

KL and the partition function

Suppose $P(X_1,\ldots,X_n) = Z^{-1} \prod_i \Psi_i(C_i)$ is Markov Network

Theorem: For any Q (not necessarily fully factorized)

$$\ln Z = F[P;Q] + D(Q||P)$$

Hereby, F[P;Q] is the following energy functional

$$F[P;Q] = \sum_i \mathbb{E}_Q[\ln \Psi_i] + H(Q)$$

Reverse KL vs. log-partition function

$$\ln Z = F[P;Q] + D(Q||P) \qquad F[P;Q] = \sum_i \mathbb{E}[\ln \Psi_i] + H(Q)$$

$D(Q||P) \geq 0$

Maximizing energy functional ⇔ Minimizing reverse KL

Corollary:

Energy function is lower bound on log partition function

$$P(x) = \frac{1}{Z} \prod_i \psi_i(c_i)$$

$$P(x) \leq \frac{1}{F(P;Q)} \prod_i \psi_i(c_i)$$

Implies upper bound on event probabilities!

Optimizing for mean field approximation

- Want to solve (*) $\max_Q F[P;Q] = \max_Q \sum_j \mathbb{E}_Q[\ln \Psi_j] + \sum_i H(Q_i)$

$$\text{s.t.} \sum_{x_i} Q_i(x_i) = 1$$

- Solved via Lagrange multipliers: there exist $\lambda_1 \dots \lambda_n$ s.t. optimization of (*) is equivalent to

$$\max_Q \sum_j \mathbb{E}_Q[\ln \psi_j] + \sum_j H(Q_j) + \sum_j \lambda_j \left[\sum_{x_j} Q_j(x_j) - 1\right]$$

Differentiate and set to 0!

Minimum, Maximum or saddle point

Theorem: Q stationary point iff for each i and x_i:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left(\sum_j \mathbb{E}[\ln \Psi_j \mid x_i]\right)$$

Fixed point iteration for MF

- Initialize factors $Q^{(0)}_i$ arbitrarily; t=0
- Until converged, do
  - t← t+1
  - For each variable i and each assignment x_i do

$$Q_i(x_i)^{(t+1)} = \frac{1}{Z_i} \exp\Big(\sum_j \mathbb{E}_{Q^{(t)}}[\ln \Psi_j \mid x_i]\Big)$$

$$Z_i = \sum_{x_i} Q_i(x_i)$$

- Guaranteed to converge! ☺
- Gives both approx. distribution Q and lower bound on ln Z
- Can get stuck in local optimum ☹

Computing updates

Need to compute

$$Q_i(x_i)^{(t+1)} = \frac{1}{Z_i} \exp\Big(\sum_j \mathbb{E}_{Q^{(t)}}[\ln \Psi_j \mid x_i]\Big)$$

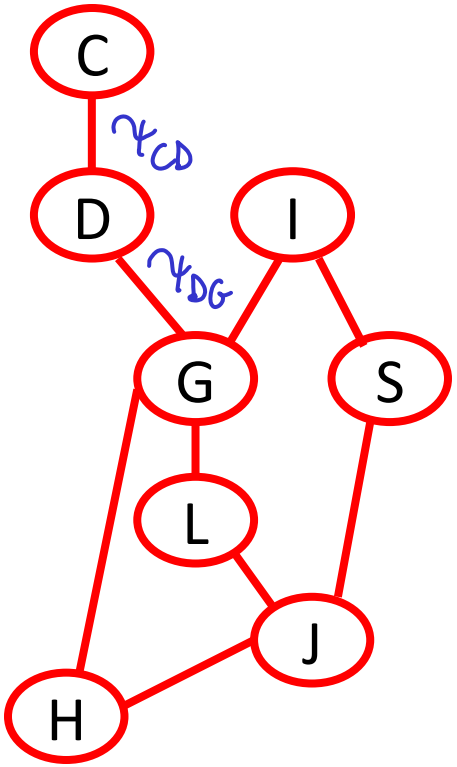
Must compute expected log potentials: $(*) = \mathbb{E}_Q[\ln \Psi_j \mid x_i]$

$$(*) = \sum_x Q^{(t)}(x \mid x_i) \ln \Psi_j(x) = \sum_{c_j \sim x_i} Q^{(t)}(c_j \mid x_i) \ln \Psi_j(c_j)$$

$= \Psi_j(c_j)$

$= \prod_{k \in c_j \setminus i} Q_k(x_k)$

Example iteration



$$\mathbb{E}_Q[\ln \psi_{DG} \mid X_D = 1]$$

$$= \sum_{X_G} \underbrace{Q(X_G \mid X_D = 1)}_{= Q_G(X_G)} \ln \psi_{DG}(X_D = 1, X_G)$$

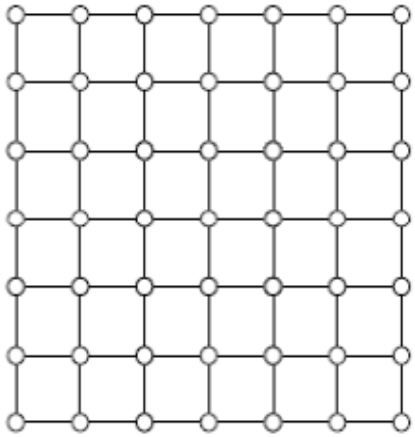
$$Q_D(X_D = 1) = \frac{1}{Z_D} \exp\left(\mathbb{E}_Q[\ln \psi_{DG} \mid X_D = 1] + \mathbb{E}_Q[\ln \psi_{CD} \mid X_D = 1]\right)$$

Structured mean field

Goal of variational inference:

Approximate complex distribution by simple distribution

True dist.

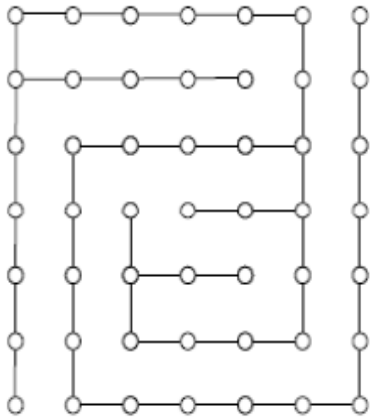


$$p(x) \propto \prod_c \phi_c(x_c)$$

Fully-factorized mean field

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

Structured mean field



$$q(x) \propto q_A(x_A)\, q_B(x_B)$$

Structured mean-field approximations

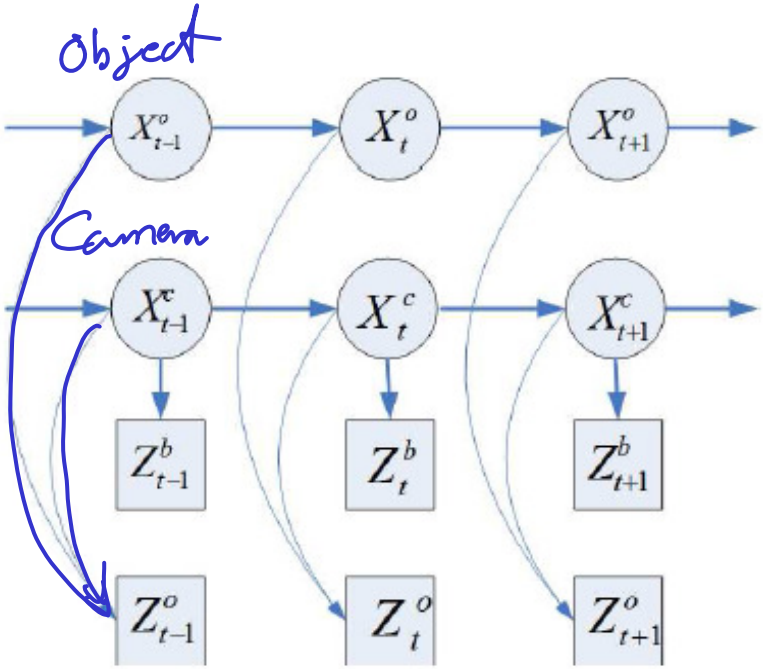
- Can get better approximations using **structured approximations**:

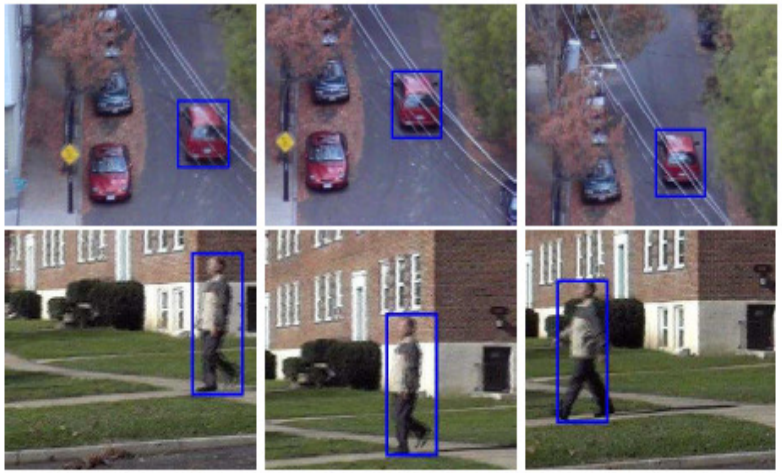
$$\max_{Q \in \mathcal{M}} F[P; Q] = \max_{Q \in \mathcal{M}} \sum_j \mathbb{E}_Q[\ln \Psi_j] + H(Q)$$

- Only need to be able to compute energy functional
- Can do whenever we can perform efficient inference in Q (e.g., chains, trees, low-treewidth models)
  - Update equations look similar as for fully-factorized case (see reading)

Example: Factorial HMM

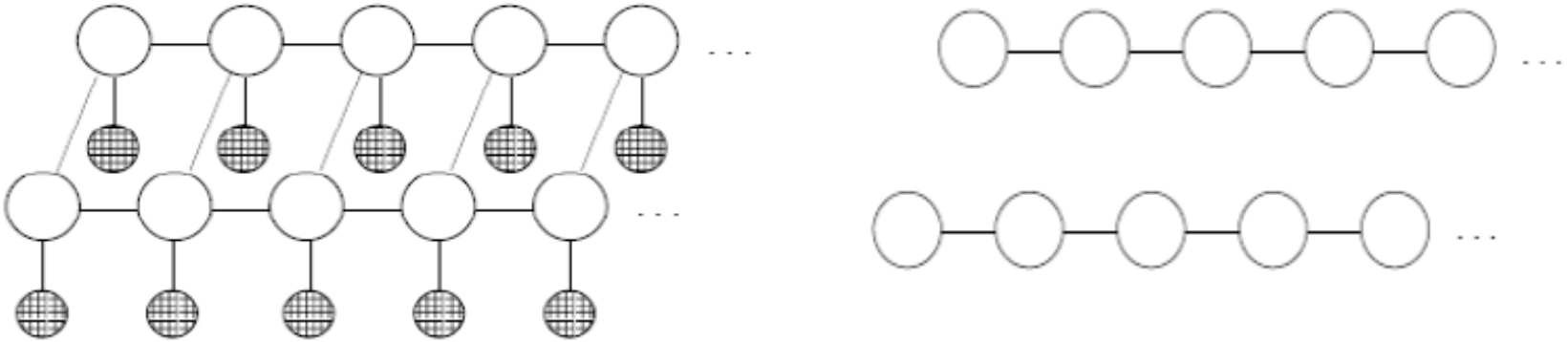
- Simultaneous tracking and camera registration
- State space decomposed into object location and camera parameters





Mei and Porikli '08

Variational approximations for FHMMs



$$\max_{Q \in \mathcal{M}} F[P;Q] = \max_{Q \in \mathcal{M}} \sum_j \mathbb{E}_Q[\ln \Psi_j] + H(Q)$$

- Approximate posterior by independent chains

$$\mathcal{M} = \left\{ Q : Q(\mathbf{X}) = \prod_c \prod_t Q_{c,t}(X_{c,t} | X_{c,t-1}) \right\}$$

Summary: Variational inference

- Approximate complex (intractable) distribution by simpler distribution that is "as close as possible"
- **Simple** = tractable (efficient inference)
- **Closeness** = Reverse KL (efficient to compute)
- Interpretation: Optimize **lower bound** on the log-partition function
  - Implies upper bound on event probabilities
- Efficient algorithm that's guaranteed to converge (in contrast to Loopy BP..), but possibly to local optimum

Approximate inference

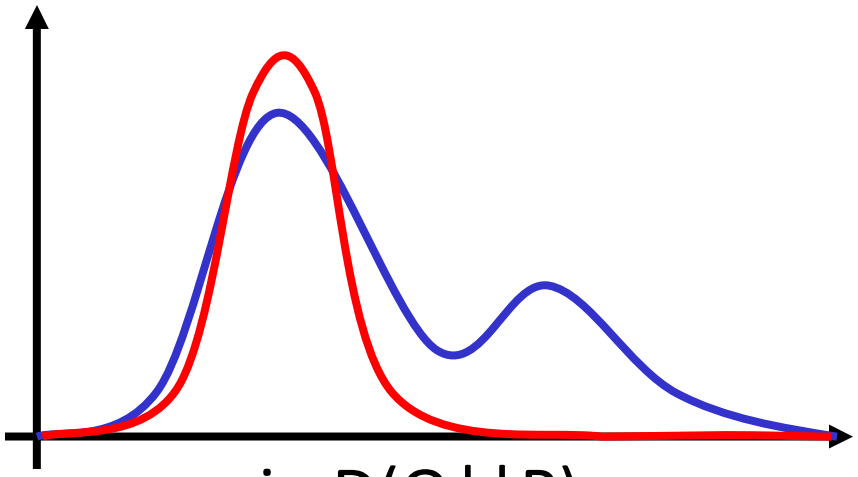
- Three major classes of general-purpose approaches
- **Message passing**
  - E.g.: Loopy Belief Propagation (today!)
- **Inference as optimization**
  - Approximate posterior distribution by simple distribution
  - Mean field / structured mean field
  - Assumed density filtering / expectation propagation
- **Sampling based inference**
  - Importance sampling, particle filtering
  - Gibbs sampling, MCMC
- Many other alternatives (often for special cases)

KL-divergence the "right" way:

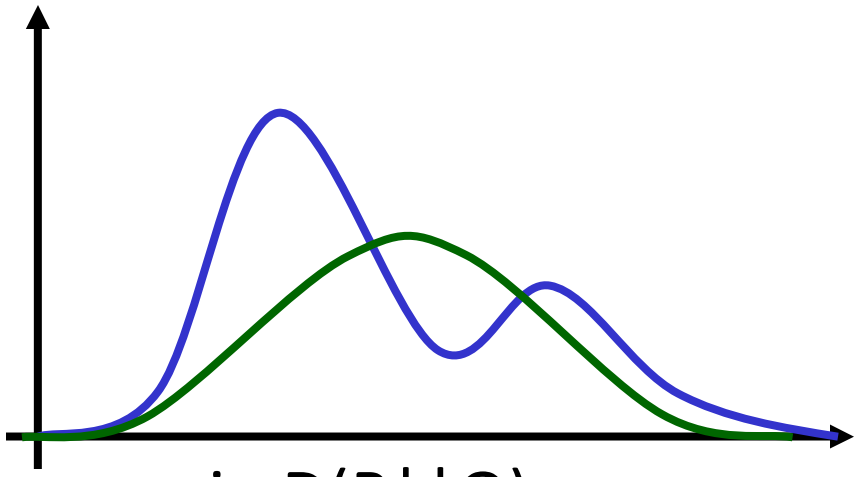
- Find distribution Q* $\in$ M:

$$Q^* = \underset{Q \in \mathcal{M}}{\operatorname{argmin}} D(P||Q)$$

- In some applications, can compute D(P || Q)
  - Important example: Assumed density filtering in DBNs



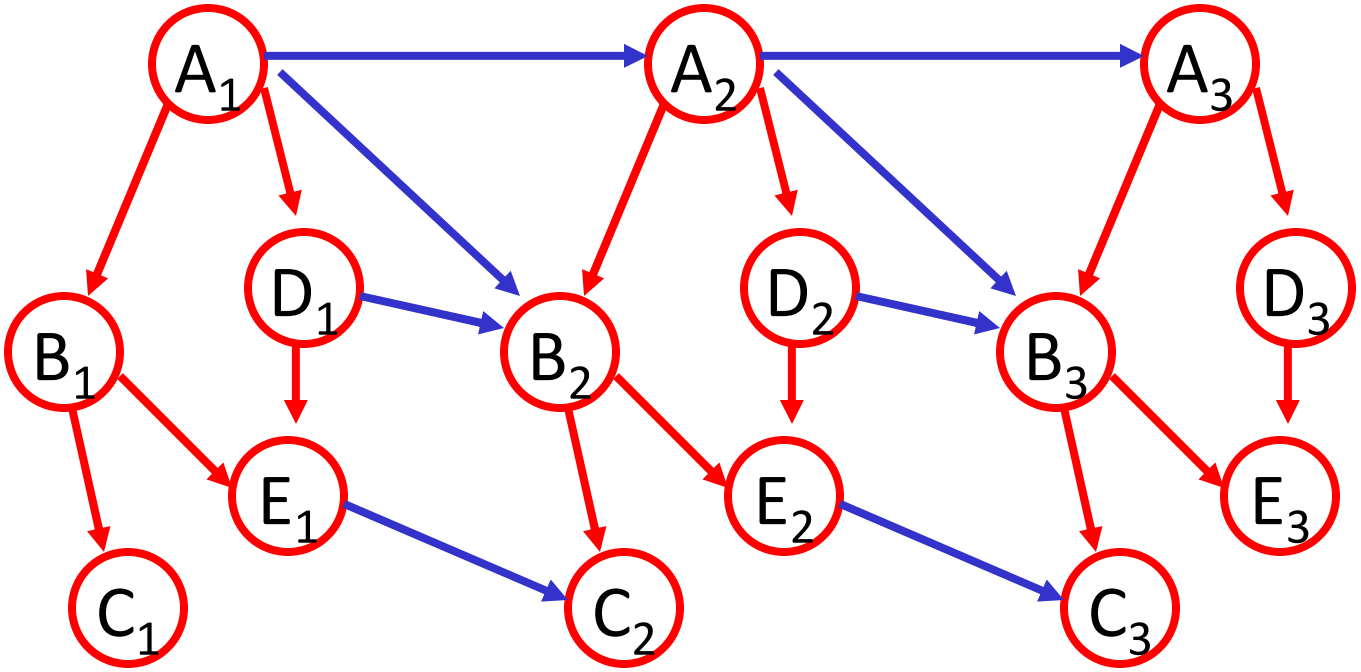
min D(Q||P)



min D(P||Q)

Recall: Dynamic Bayesian Networks

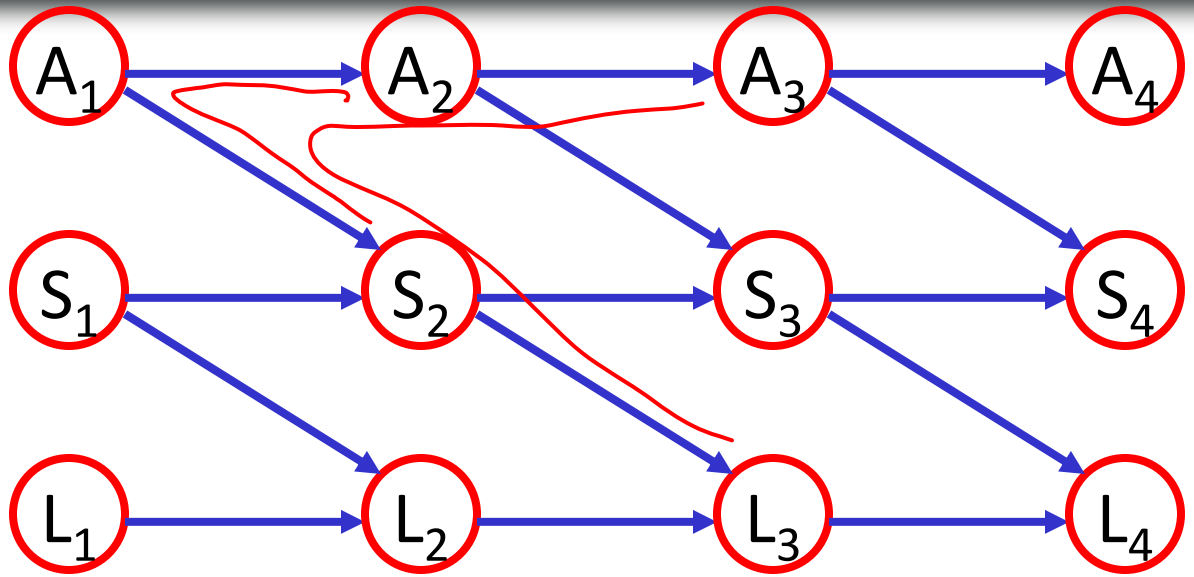
- At every timestep have a Bayesian Network



$S_t = \{ A_t, B_t, \ldots, E_t, S \}$

- Variables at each time step t called a "slice" $\mathbf{S_t}$
- "Temporal" edges connecting S_{t+1} with S_t

Flow of influence in DBNs



acceleration

speed

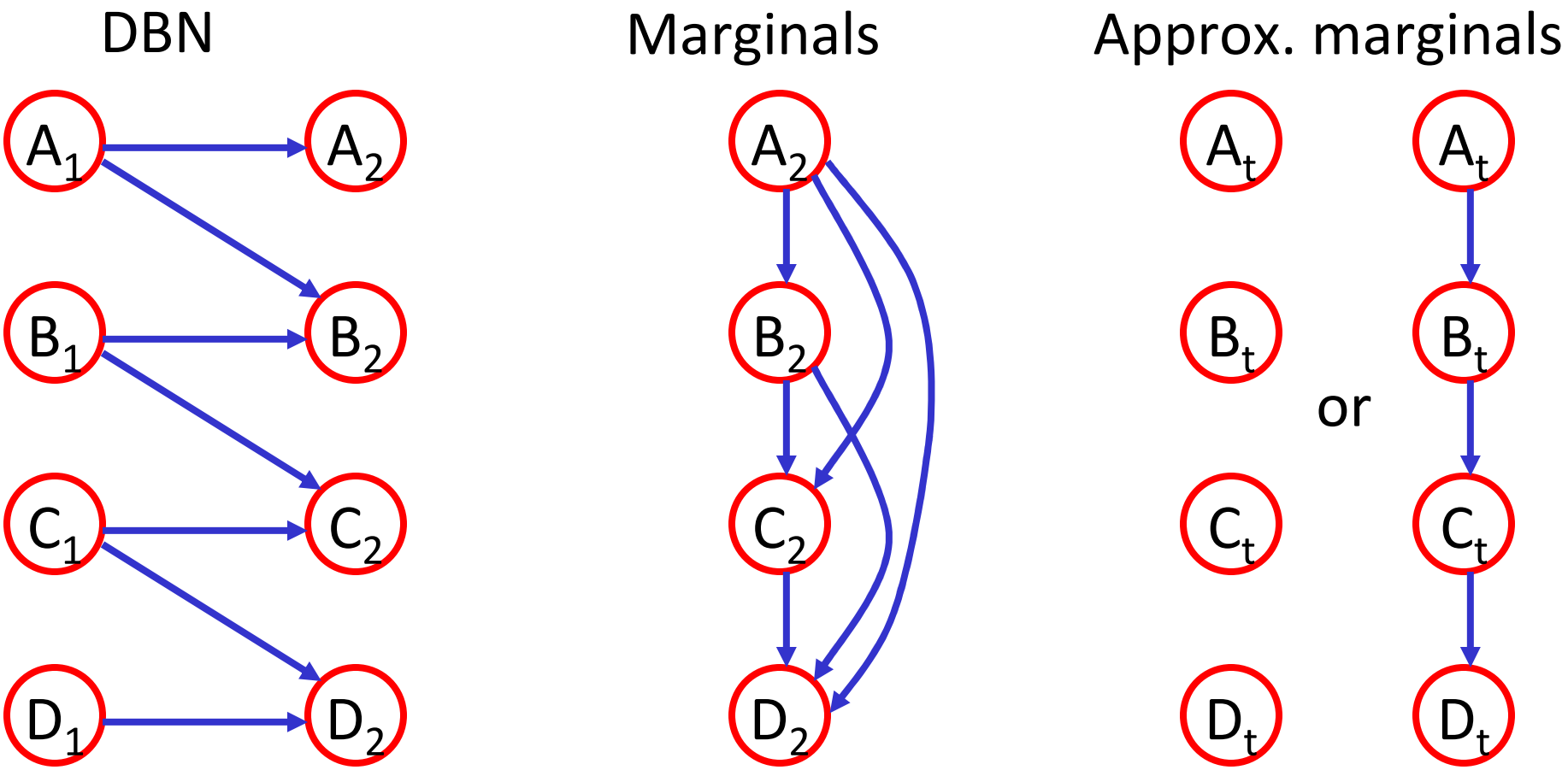
location

$A_1 \perp S_1$ ✓ $A_2 \perp S_2$ ✗

$A_1 \perp L_1$ ✓ $A_2 \perp L_2$ ✓ $A_3 \perp L_3$ ✗

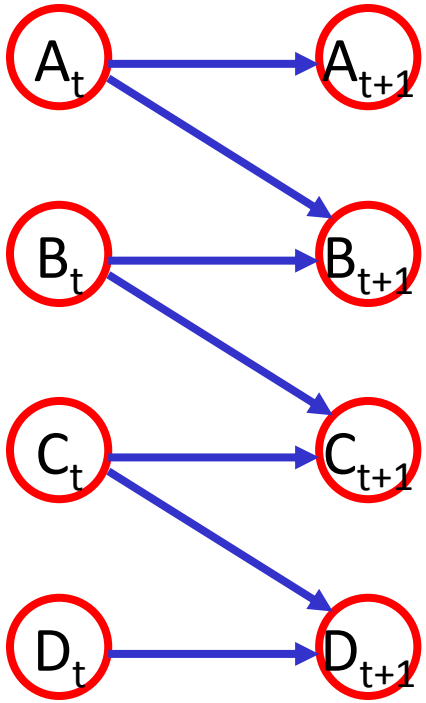
- Can we do efficient filtering in BNs?

Approximate inference in DBNs?



Want to find **tractable** approximation to marginals that's **as close** to true marginals as possible

Assumed Density Filtering



$$D(P(S_{t+1}) \,||\, Q(S_{t+1}))$$

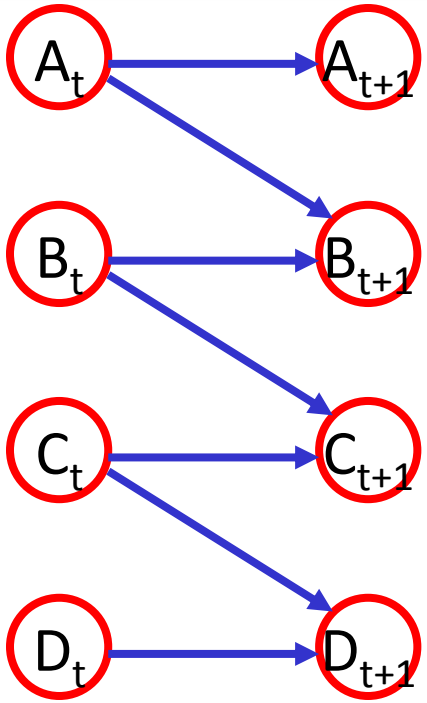
$$= \sum_{S_{t+1}} P(S_{t+1}) \log \frac{P(S_{t+1})}{Q(S_{t+1})}$$

$$= \text{const} - \underbrace{\sum_{S_{t+1}} P(S_{t+1}) \log Q(S_{t+1})}_{(*)}$$

- Assume distribution $P(\mathbf{S_t})$ for slice t factorizes
- $P(\mathbf{S_{t+1}})$ is fully connected ☹
- Want to compute best-approximation Q* for $P(\mathbf{S_{t+1}})$

$$Q^* = \text{argmin } D(P \,||\, Q)$$

Assumed Density Filtering



$$\sum_{S_{t+1}} P(S_{t+1}) \log \underbrace{Q(S_{t+1})}_{= \prod_i Q_i(S_{i,t+1})}$$

$$= \sum_i \sum_{S_{t+1}} P(S_{t+1}) \log Q_i(S_{i,t+1})$$

$$= \sum_i \sum_{S_{i,t+1}} P(S_{i,t+1}) \log Q_i(S_{i,t+1})$$

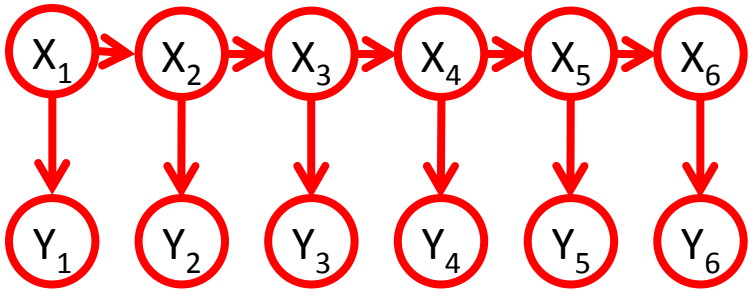
E.g., $P(A_{t+1})$

Can compute expectations efficiently

Get optimal Q^* by setting $Q_i^*(S_{i,t+1}) = P(S_{i,t+1})$

Recall: Bayesian filtering

- Start with $P(X_1)$
- At time t
  - Assume we have $P(X_t \mid y_{1\ldots t-1})$
  - Condition: $P(X_t \mid y_{1\ldots t})$



$$P(X_t \mid y_{1\ldots t}) \propto P(X_t \mid y_{1\ldots t-1})\, P(y_t \mid X_t, y_{1\ldots t-1})$$

cond. ind. $P(y_t \mid X_t)$

  - Prediction: $P(X_{t+1}, X_t \mid y_{1\ldots t})$

$$P(X_{t+1}, X_t \mid y_{1\ldots t}) = P(X_t \mid y_{1\ldots t}) \cdot P(X_{t+1} \mid X_t, y_{1\ldots t})$$

$= P(X_{t+1} \mid X_t)$

  - Marginalization: $P(X_{t+1} \mid y_{1\ldots t})$

$$P(X_{t+1} \mid y_{1\ldots t}) = \sum_{X_t} P(X_{t+1}, X_t \mid y_{1\ldots t})$$

Assumed Density Filtering

- Start with $P(\mathbf{S}_1)$
- At every time step t: tractable approximation Q_t
  $Q_t(\mathbf{S}_t) \approx P(\mathbf{S}_t \mid \mathbf{O}_{1:t-1})$
- **Condition** on observation $\mathbf{O}_t \subseteq \mathbf{S}_t$: $Q_t(\mathbf{S}_t \mid \mathbf{O}_t)$
- **Predict**: multiply transition model to get $Q_t(\mathbf{S}_{t+1},\mathbf{S}_t \mid \mathbf{O}_t)$

  $$Q_t(\mathbf{S}_{t+1},\mathbf{S}_t \mid \mathbf{O}_t) = Q_t(\mathbf{S}_t \mid \mathbf{O}_t)\, P(\mathbf{S}_{t+1} \mid \mathbf{S}_t)$$
- **Marginalize $\mathbf{S}_t$**
  - This is intractable (connects all variables in $\mathbf{S}_{t+1}$)
  - Approximate $Q_t(\mathbf{S}_{t+1} \mid \mathbf{O}_t)$ by Q* s.t.
    $Q^* = \text{argmin}_Q\, D(Q_t(\mathbf{S}_{t+1}) \,||\, Q(\mathbf{S}_{t+1}))$
  - This is done by matching moments:
    for discrete models, ensure that $Q_{t+1}(\mathbf{s}_{t+1}) = Q_t(\mathbf{s}_{t+1} \mid \mathbf{o}_t)$

Summary of Assumed Density Filtering

- Variational inference technique for dynamical Bayesian Networks
- Find tractable approximation for each time slice that minimizes KL divergence (in the “right” way)
- Can show that errors don’t add up too much
- Examples:
  - Tractable inference in DBNs
  - Unscented Kalman Filter

Summary: Inference as optimization

- Approximate intractable distribution by a **tractable** one
- **Optimize parameters** of the distribution to make approximation as tight as possible
- Common distance measure: KL-divergence (both ways)
  - Special case of α-divergence
- Can get upper bounds on event probabilities, etc.