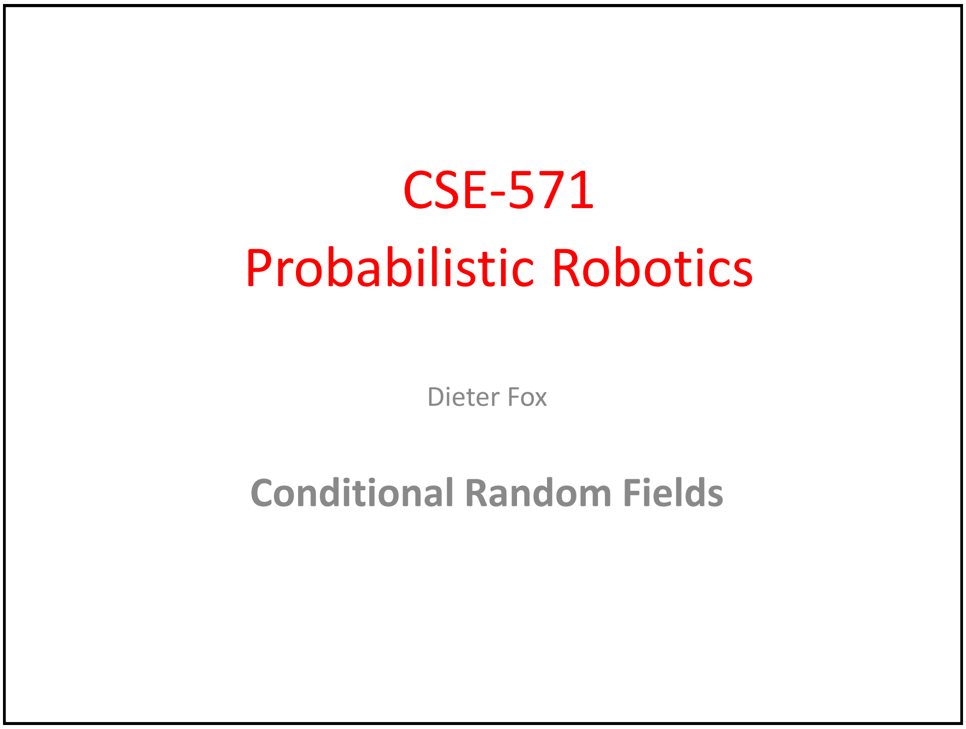
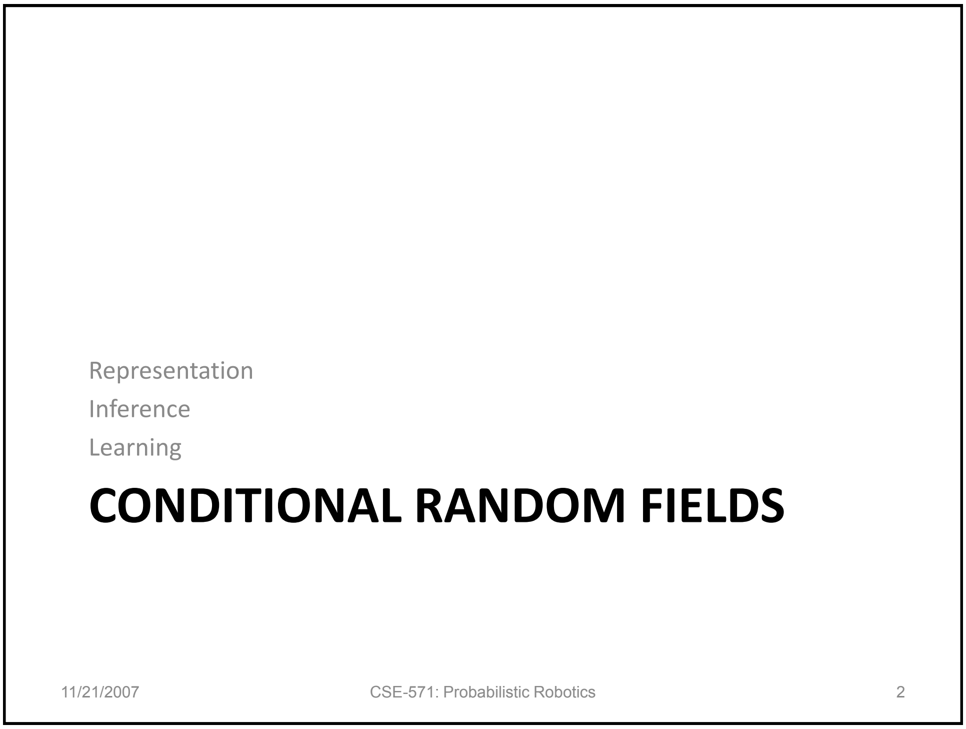


CSE-571
Probabilistic Robotics

Dieter Fox

Conditional Random Fields



Representation
Inference
Learning

CONDITIONAL RANDOM FIELDS

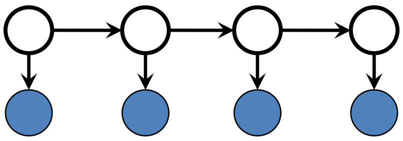
Conditional Random Fields

- Discriminative, undirected graphical model
- Introduced for labeling sequence data to overcome weaknesses of Hidden Markov Models [Lafferty-McCallum-Pereira: ICML-01]
- Applied successfully to
  - Natural language processing [McCallum-Li: CoNLL-03], [Roth-Yih: ICML-05]
  - Social network analysis [McCallum-CorradaEmmanuel-Wang: IJCAI-05]
  - Computer vision [Kumar-Hebert: NIPS-04], [Quattoni-Collins-Darrel: NIPS-05]
  - Activity recognition [Liao-Fox-Kautz: IJRR-07, Smimchisescu-Kanaujia-Li-Metaxus: ICCV-05]

Hidden Markov Models

Hidden states **x**

Observations **z**

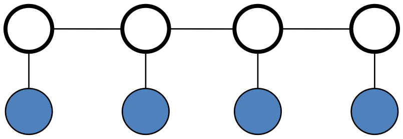


- Directed graphical model

$$p(\mathbf{x}_{0:K}, \mathbf{z}_{1:K}) = \prod_{k=} \mathbf{x} \quad \mathbf{x} \quad (\mathbf{z}_k \mid \mathbf{x}_k)$$

$$p(\mathbf{x}_{0:K} \mid \mathbf{z}_{1:K}) = \prod_{k=} \mathbf{x} \quad \mathbf{x} \quad (\mathbf{z}_k \mid \mathbf{x}_k)$$

Conditional Random Fields



Hidden states **x**

Observations **z**

- Directly models conditional probability p(x|z)

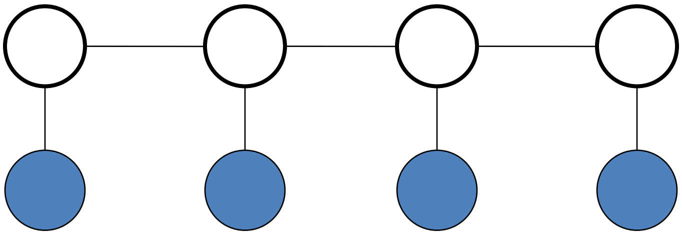
(instead of modeling p(z|x) and p(x), and using Bayes rule to infer p(x|z)).

- No independence assumption on observations needed!

Conditional Probability

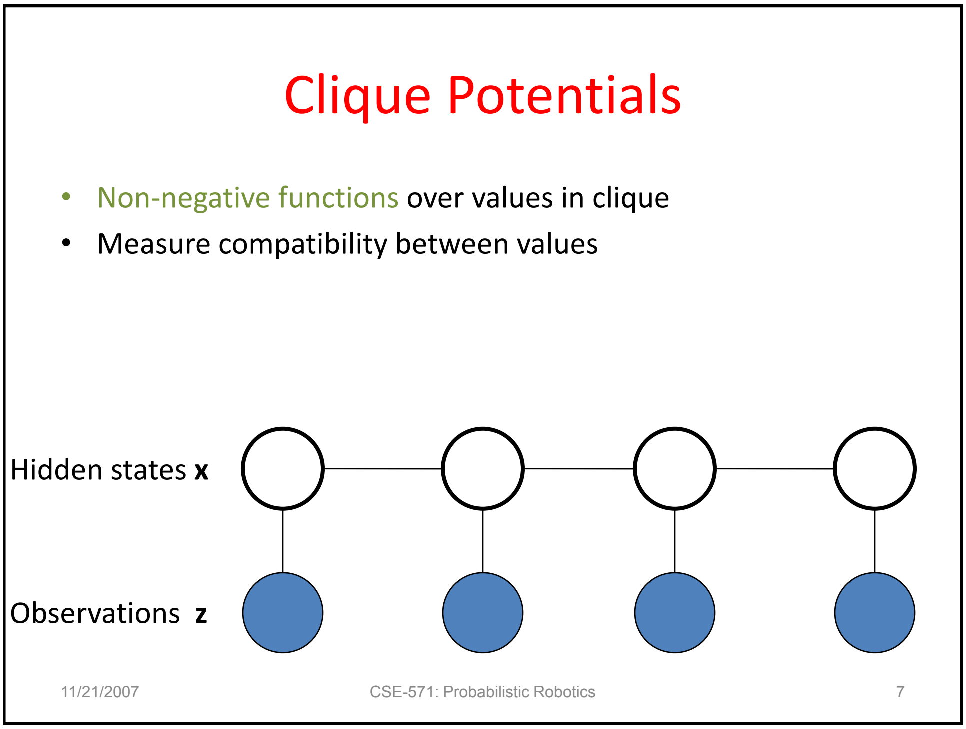
- Conditional probability factorizes into **clique potentials**:

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \in} \quad \mathbf{x} \quad \mathbf{z}$$



Hidden states **x**

Observations **z**

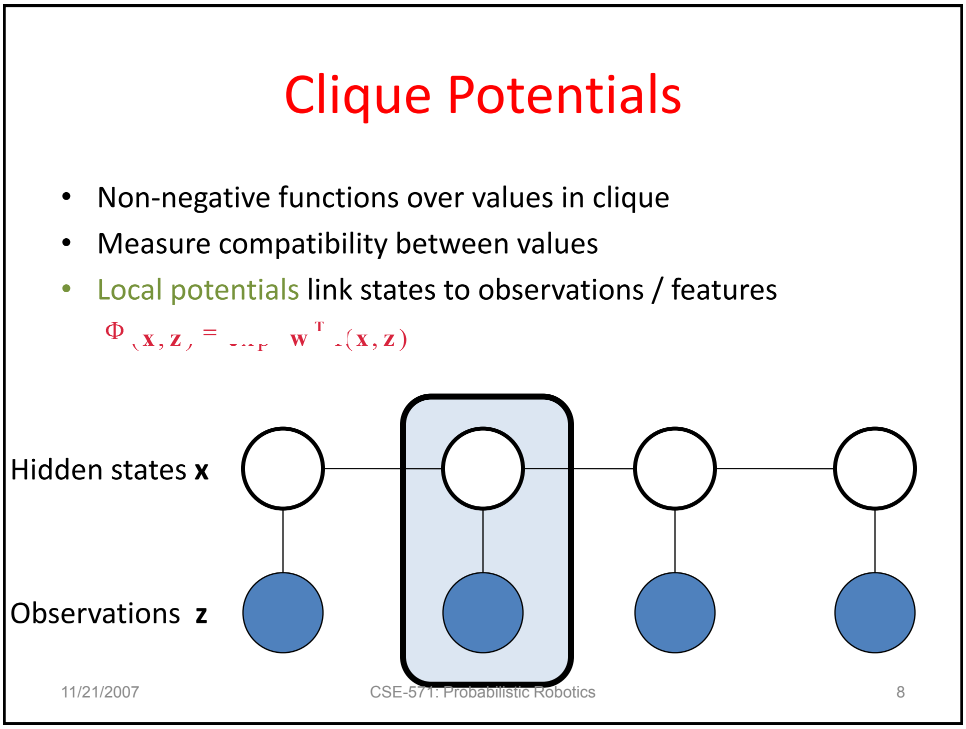


Clique Potentials

- Non-negative functions over values in clique
- Measure compatibility between values

Hidden states **x**

Observations **z**



Clique Potentials

- Non-negative functions over values in clique
- Measure compatibility between values
- Local potentials link states to observations / features

$\Phi(\mathbf{x},\mathbf{z}) = \exp(\mathbf{w}^T \mathbf{f}(\mathbf{x},\mathbf{z}))$

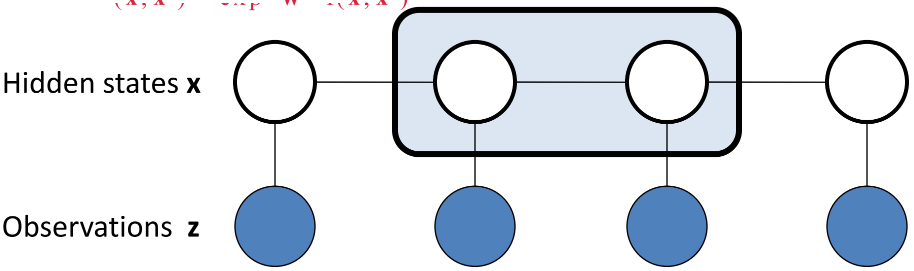
Hidden states **x**

Observations **z**

Clique Potentials

- Non-negative functions over values in clique
- Measure compatibility between values
- Local potentials link states to observations / features
- Neighborhood potentials link states to neighboring states

$\Phi\ (\mathbf{x},\mathbf{x}\ ) = \exp\ \mathbf{w}^{T}\ (\mathbf{x},\mathbf{x}')$



Conditional Distribution Revisited

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c\in} \quad \mathbf{x} \quad \mathbf{z}$$

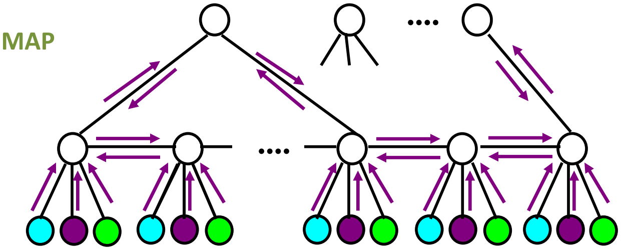
$$= \frac{1}{Z(\mathbf{z})} \exp\left\{ \sum \mathbf{w} \quad \mathbf{x} \quad \mathbf{z} \right\}$$

- Normalizer can grow exponentially in number of variables:

$$Z(\mathbf{z}) = \sum_{\mathbf{x}} \quad \left\{ \sum \mathbf{w} \quad \mathbf{x} \quad \mathbf{z} \right\}$$

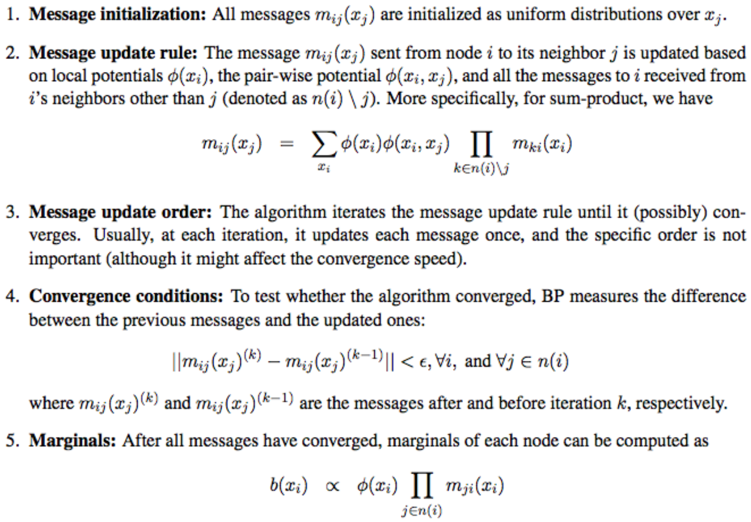
Inference via Belief Propagation

- BP computes posteriors via local message passing
  - Sum-product for posterior
  - Max-product for MAP



- Exact if network has no loops
- Otherwise, run loopy belief propagation and hope it works

Sum-Product Belief Propagation



1. **Message initialization:** All messages $m_{ij}(x_j)$ are initialized as uniform distributions over x_j.

2. **Message update rule:** The message $m_{ij}(x_j)$ sent from node i to its neighbor j is updated based on local potentials $\phi(x_i)$, the pair-wise potential $\phi(x_i, x_j)$, and all the messages to i received from i's neighbors other than j (denoted as $n(i) \setminus j$). More specifically, for sum-product, we have

$$m_{ij}(x_j) = \sum_{x_i} \phi(x_i)\phi(x_i, x_j) \prod_{k \in n(i) \setminus j} m_{ki}(x_i)$$

3. **Message update order:** The algorithm iterates the message update rule until it (possibly) converges. Usually, at each iteration, it updates each message once, and the specific order is not important (although it might affect the convergence speed).

4. **Convergence conditions:** To test whether the algorithm converged, BP measures the difference between the previous messages and the updated ones:

$$||m_{ij}(x_j)^{(k)} - m_{ij}(x_j)^{(k-1)}|| < \epsilon, \forall i, \text{ and } \forall j \in n(i)$$

where $m_{ij}(x_j)^{(k)}$ and $m_{ij}(x_j)^{(k-1)}$ are the messages after and before iteration k, respectively.

5. **Marginals:** After all messages have converged, marginals of each node can be computed as

$$b(x_i) \propto \phi(x_i) \prod_{j \in n(i)} m_{ji}(x_i)$$

Inference via Gibbs Sampling

- Basic Markov chain Monte Carlo technique
- Goal: generate sequence of samples drawn from posterior

$$p(\mathbf{x} \mid \mathbf{z}) \propto \exp\ \mathbf{w}\ \mathbf{f}(\mathbf{x}, \mathbf{z})$$

- Initialize all x_k to a random value
- At each step pick an x_k and sample from the conditional:

$$p(x_k \mid \mathbf{x}_{-k}\ \mathbf{z}) \propto \exp\ \mathbf{w}\ \mathbf{f}(\mathbf{x}, \mathbf{z})$$

- Problem: difficult to move between modes of posterior
- Many alternatives: block sampling, slice sampling, MC-SAT

Parameter Learning

- Conditional distribution parameterized via weights **w**:

$$p(\mathbf{x} \mid \mathbf{z}, \mathbf{w}) = \frac{1}{Z(\mathbf{z}, \mathbf{w})} \exp\left\{ \sum \mathbf{w}\ \ \mathbf{x}\ \ \mathbf{z} \right\}$$

- Maximize **conditional log-likelihood** with **shrinkage prior**:

$$\mathrm{L}(\mathbf{w}) = \log p(\mathbf{x} \mid \mathbf{z}, \mathbf{w}) - \frac{\mathbf{w}^{\mathrm{T}}\mathbf{w}}{\sigma}$$

- No closed-form solution, gradient requires inference:

$$\nabla L(\mathbf{w}) = \mathbf{f}(\mathbf{x}, \mathbf{z}) - E_{p(\mathbf{x}' \mid \mathbf{z}, \mathbf{w})}\, \mathbf{f}(\mathbf{x}', \mathbf{z}) - \frac{\mathbf{w}}{\sigma}$$

- Maximization via stochastic gradient, L-BFGS, conjugate gradient

Pseudo-Likelihood Learning

- Alternative: maximize **pseudo log-likelihood** [Besag: 1975]

$$PL(\mathbf{w}) = \sum_{i} \text{[illegible]}\, \mathbf{x}_{i} \,\text{[illegible]}\, (\mathbf{x}_{i}), \mathbf{w}) - \frac{\mathbf{w}^{T}\mathbf{w}}{\sigma}$$

- Gradient computation does not require inference

- Very efficient, works surprisingly well in practice

[Kumar-Hebert: ICCV-03], [Richardson-Domingos: ML-04], [Liao-Fox-Kautz: IJRR-07]

Conclusions

- Graphical models provide powerful and flexible framework for learning and reasoning about complex relationships
- Conditional Random Fields
  - Can handle high-dimensional features
  - No need to worry about dependencies