Graphical Models and Conditional Random Fields

Presenter: Shih-Hsiang Lin

Bishop, C. M., “Pattern Recognition and Machine Learning,” Springer, 2006
Sutton, C., McCallum, A., “An Introduction to Conditional Random Fields for Relational Learning,” Introduction to Statistical Relational Learning, MIT Press. 2007
Rahul Gupta, “Conditional Random Fields,” Dept. of Computer Science and Engg., IIT Bombay, India.

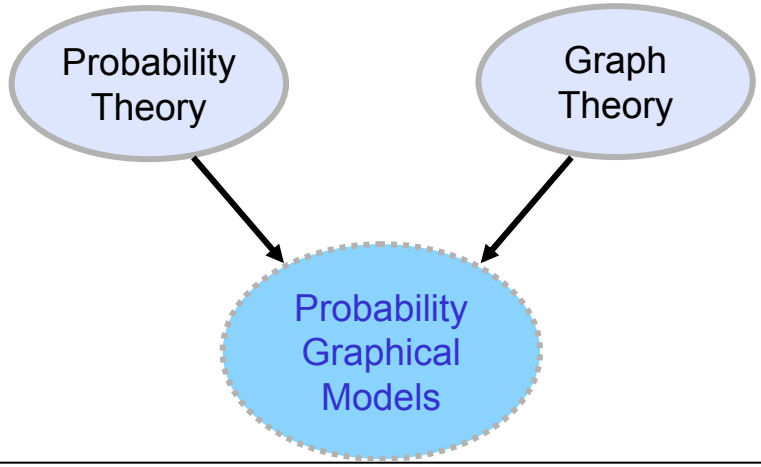
Overview

- Introduction to graphical models
- Applications of graphical models
- More detail on conditional random fields
- Conclusions

Introduction to Graphical Models

Power of Probabilistic Graphical Models

- Why do we need graphical models
  - Graphs are an intuitive way of representing and visualizing the relationships between many variables
    - Used to design and motivate new models
  - A graph allows us to abstract out the conditional independence relationships between the variables from the details of their parametric forms.
    - Provide a new insights into existing model
  - Graphical models allow us to define general message-passing algorithms that implement probabilistic inference efficiently
    - Graph based algorithms for calculation and computation



Probability Theory

- What do we need to know in advance
  - Probability Theory
    - Sum Rule (Law of Total Probability or Marginal Probability)

$$p(x) = \sum_{y} p(x, y)$$

    - Product Rule (Chain Rule)

$$p(x, y) = p(x \mid y)p(y) = p(y \mid x)p(x)$$

    - From the above we can derive Bayes' theorem

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} = \frac{p(x \mid y)p(y)}{\sum_{y} p(x \mid y)p(y)}$$

Conditional Independence and Marginal Independence

- Conditional Independence

$$x \perp\!\!\!\perp \mathrm{y} \mid \mathrm{z} \quad \Leftrightarrow \quad p(x \mid y, z) = p(x \mid z)$$

which is equivalent to saying

$$p(x, y \mid z) = p(x \mid y, z)p(y \mid z) = p(x \mid z)p(y \mid z)$$

  - Conditional independence crucial in practical applications since we can rarely work with a general joint distribution

- Marginal Independence

$$x \perp\!\!\!\perp \mathrm{y} \quad \Leftrightarrow \quad a \not\perp\!\!\!\perp \mathrm{b} \mid \emptyset \quad \Leftrightarrow \quad p(x, y) = p(x)p(y)$$

empty set

- Example
  - Amount of Speeding Fine $\perp\!\!\!\perp$ Type of Car | Speed
  - Lung Cancer $\perp\!\!\!\perp$ Yellow Teeth | Smoking
  - Child's Genes $\perp\!\!\!\perp$ Grandparents' Genes | Parents' Genes
  - Ability of Team A $\perp\!\!\!\perp$ Ability of Team B

Graphical models

Directed Graph Undirected Graph

- A graphical model comprises nodes connected by links
  - Nodes (vertices) correspond to random variables
  - Links (edges or arcs) represents the relationships between the variables
- Directed graphs are useful for expressing casual relationships between random variables
- Undirected graphs are better suited to expressing soft constraints between random variables

Directed Graphs

- Consider an arbitrary distribution $p(a,b,c)$, we can write the joint distribution in the form
  - By successive application of the product rule

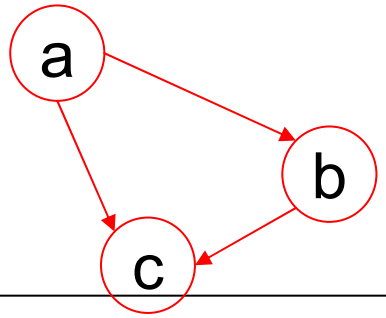
$$p(a,b,c) = p(c \mid a,b)p(a,b)$$

or

$$p(a,b,c) = p(c \mid a,b)p(b \mid a)p(a)$$

* Note that this decomposition holds for any choice of joint distribution

- We then can represent the above equation in terms of a simple graphical models
  - First, we introduce a node for each of the random variables
  - Second, for each conditional distribution we add directed links to the graph

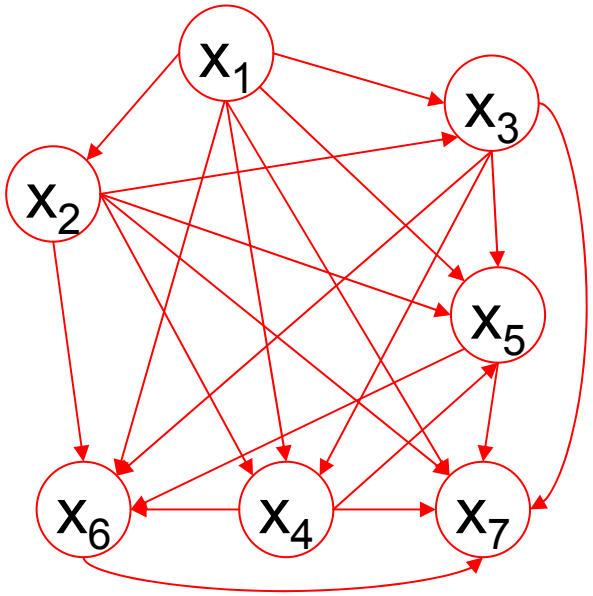


A fully connected graph

Directed Graphs (cont.)

- Let us consider another case $p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$

$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = p(x_7 \mid x_1, \cdots, x_6) \cdots p(x_2 \mid x_1) p(x_1)$$



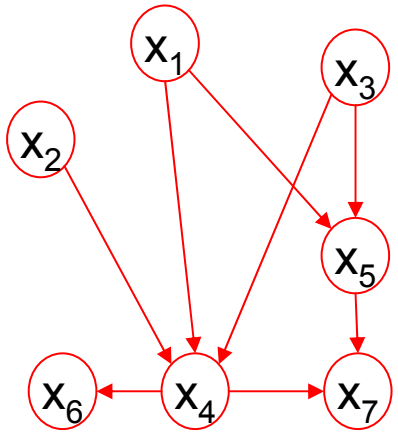
Again, it is a fully connected graph

- What would happen if some links were dropped?? (considering the relationship between nodes)

Directed Graphs (cont.)

- The joint distribution of $p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ is therefore given by

$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$
$$= p(x_1) \times p(x_2) \times p(x_3) \times p(x_4 \mid x_1, x_2, x_3) \times$$
$$p(x_5 \mid x_1, x_3) \times p(x_6 \mid x_4) \times p(x_7 \mid x_4, x_5)$$



* The joint distribution is then defined by the product of a conditional distribution for each node conditioned on the variables corresponding to the parents of that node in the graph

- Thus, for a graph with *K* nodes, the joint distribution is given by

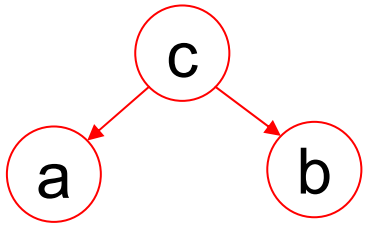
$$p(x_1, \cdots, x_K) = \prod_{k=1}^{K} p(x_k \mid parent\ (x_k))$$ where $parent\ (x_k)$ denotes the set of parents of x_k

- We always restrict the directed graph must have no directed cycles
  - Such graphs are also called directed acyclic graphs (DAGs) or Bayesian network

Directed Graph: Conditional Independence

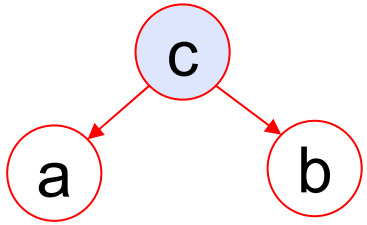
- Joint distribution over 3 variables specified by the graph

$$p(a,b,c) = p(a \mid c)p(b \mid c)p(c)$$



if node c is not observed

$$p(a,b) = \sum_c p(a \mid c)p(b \mid c)p(c) \neq p(a)p(b) \quad \Rightarrow \quad a \not\perp\!\!\!\perp b \mid \emptyset$$



if node c is observed

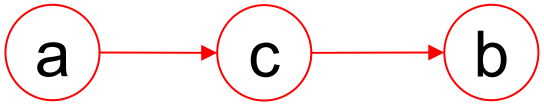
$$p(a,b \mid c) = \frac{p(a,b,c)}{p(c)} = p(a \mid c)p(b \mid c) \quad \Rightarrow \quad a \perp\!\!\!\perp b \mid c$$

The node c is said to be tail-to-tail r.w.t. this path from node a to node b
➔ this observation 'blocks' the path from a to b and cause a and b to become conditionally independent

Directed Graph: Conditional Independence (cont.)

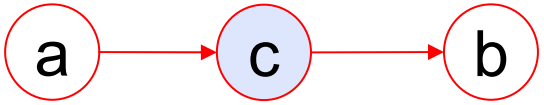
- The second example

$$p(a,b,c) = p(a)p(c \mid a)p(b \mid c)$$



if node c is not observed

$$p(a,b) = p(a)\sum_{c} p(c \mid a)p(b \mid c) = p(a)p(b \mid a) \neq p(a)p(b) \quad \Rightarrow \quad a \not\perp\!\!\!\perp b \mid \emptyset$$



if node c is observed

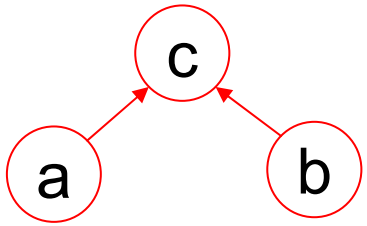
$$p(a,b \mid c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(c \mid a)p(b \mid c)}{p(c)} = p(a \mid c)p(b \mid c) \quad \Rightarrow \quad a \perp\!\!\!\perp b \mid c$$

The node c is said to be head-to-tail r.w.t. this path from node a to node b
➔ this observation 'blocks' the path from a to b and cause a and b to become conditionally independent

Directed Graph: Conditional Independence (cont.)

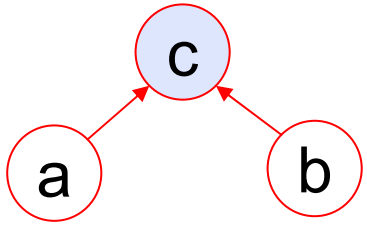
- The third example

$$p(a,b,c) = p(a)p(b)p(c \mid a,b)$$



if node c is not observed

$$p(a,b) = \sum_c p(a,b,c) = p(a)p(b)\sum_c p(c \mid a,b) = p(a)p(b) \quad \Rightarrow \quad a \perp\!\!\!\perp b \mid \emptyset$$



if node c is observed

$$p(a,b \mid c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c \mid a,b)}{p(c)} \neq p(a \mid c)p(b \mid c) \quad \Rightarrow \quad a \not\perp\!\!\!\perp b \mid c$$

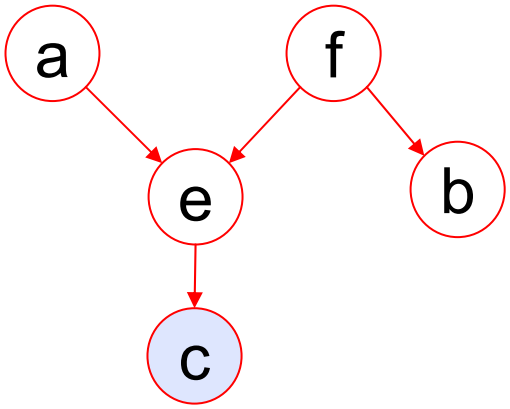
The node c is said to be head-to-head r.w.t. this path from node a to node b
➔ the conditioned node c 'unblocks' the path and renders a and b dependent

D-separation

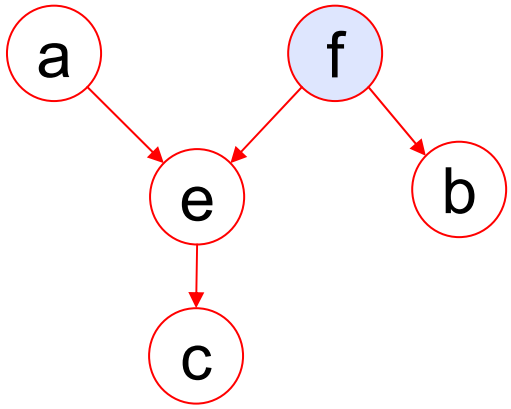
- $A \perp\!\!\!\perp B \mid C$ if C d-separated A from B
  - We need to consider all possible paths from any node in A to any node in B
  - Any such path is said to be blocked if it includes a node such that either

    (a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C

    (b) the arrows meet head-to-tail at the node, and neither the node, nor any of its descendants, is in the set C
  - If all paths are blocked, then A is said to be d-separated from B by C



$a \not\perp\!\!\!\perp b \mid c$



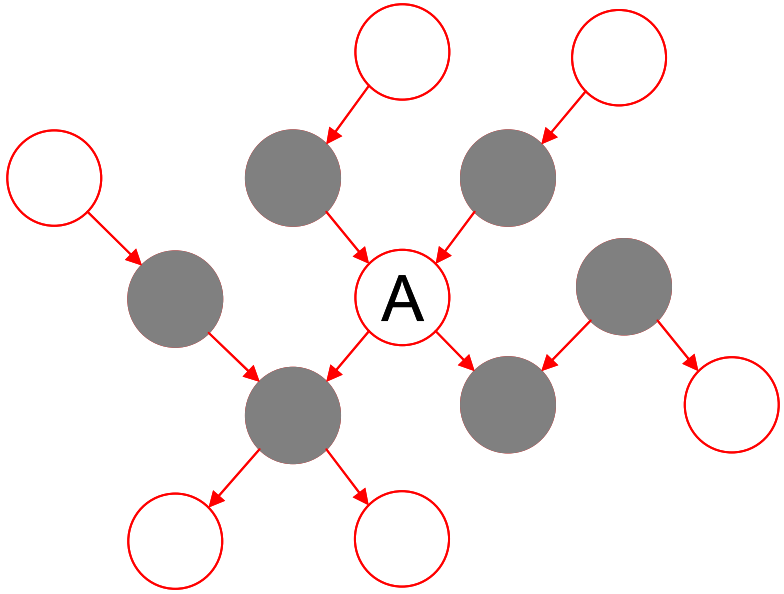
$a \perp\!\!\!\perp b \mid f$

Markov Blankets

- Markov blankets (or Markov boundary) of a node x is the minimal set of nodes that isolates nodes A from the rest of the graph
  - Every set of nodes in the network is conditionally independent of A when conditioned on the Markov blanket of the node A

$$p\left(A \mid MB\left(A\right) \cap B\right) = p\left(A \mid MB\left(A\right)\right)$$

  - MB(A)= {parents(A) and children(A) and parents-of-children(A)}

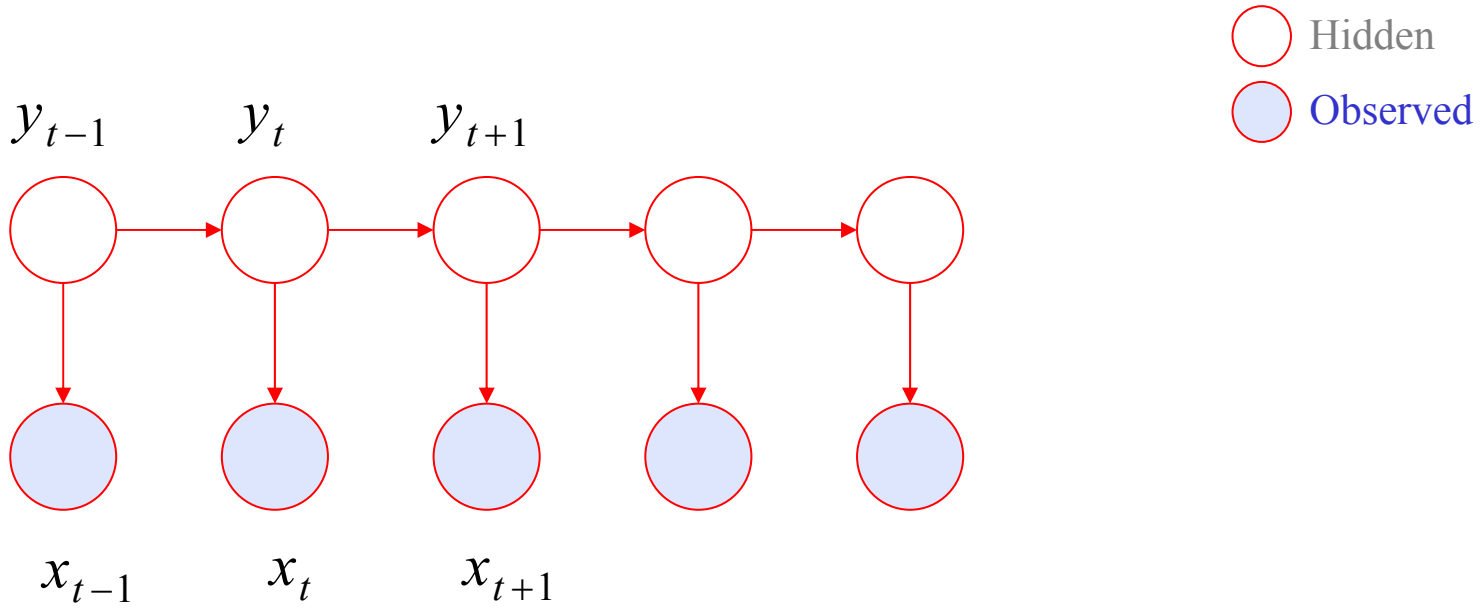


Examples of Directed Graphs

- Hidden Markov models
- Kalman filters
- Factor analysis
- Probabilistic principal component analysis
- Independent component analysis
- Mixtures of Gaussians
- Transformed component analysis
- Probabilistic expert systems
- Sigmoid belief networks
- Hierarchical mixtures of experts
- etc,…

Example: State Space Models (SSM)

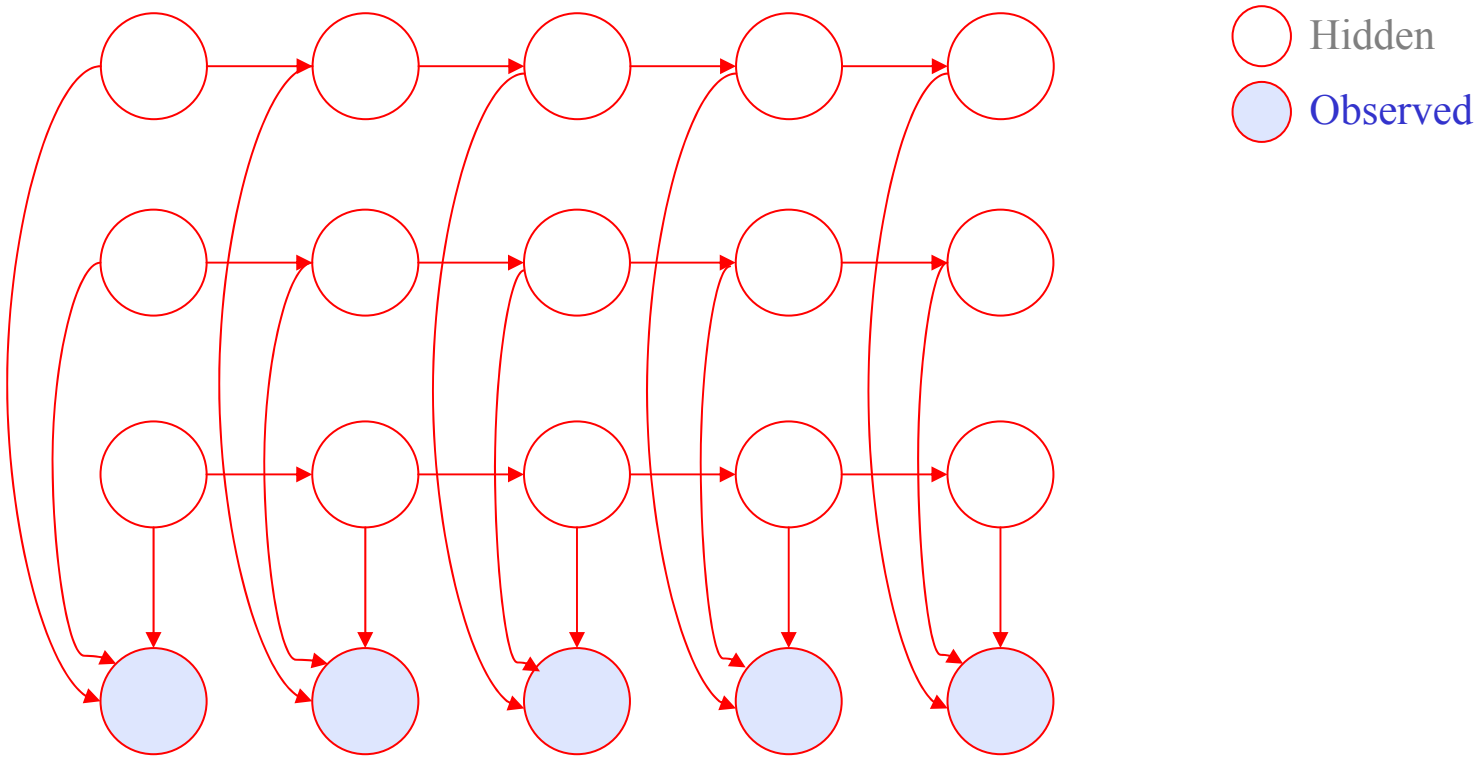
- Hidden Markov models
- Kalman filters



$$P(X,Y) = \cdots p(y_t \mid y_{t-1}) p(x_t \mid y_t) p(y_{t+1} \mid y) \cdots$$

Example: Factorial SSM

- Multiple hidden sequences

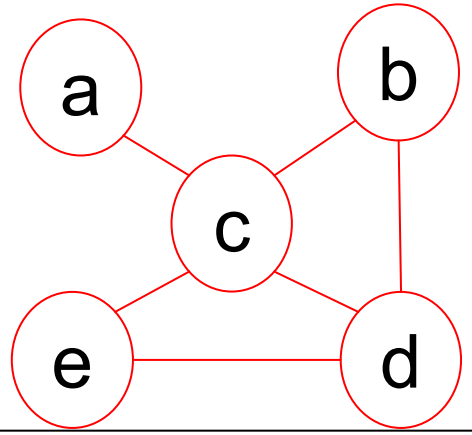


Markov Random Fields

- Random Field
  - Let $F = \{F_1, F_2, ..., F_M\}$ be a family of random variables defined on the set ***S*** , in which each random variable F_i takes a value f_i in a label set ***L***. The family F is called a random field
- Markov Random Field
  - F is said to be a Markov random field on ***S*** with respect to a neighborhood system ***N*** if and only if the following two conditions are satisfied

$$\text{Possitivity}: \quad P(f) > 0, \forall f \in F$$

$$\text{Markovianity}: \quad P(f_i \mid \text{all other } f) = P(f_i \mid neighbors(f_i))$$



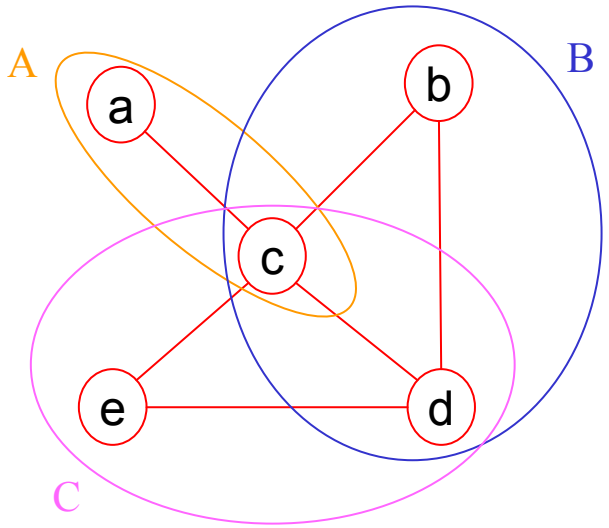
$$P(b \mid \text{all other node}) = P(b \mid c, d)$$

Undirected Graphs

- An undirected graphical model can also called Markov random fields, or also known as a Markov networks
  - It has a set of nodes each of which corresponds to a variable of group of variables, as well as a set of links each of which connects a pair of nodes
- In an undirected graphical models, the joint distribution is product of non-negative functions over the cliques of the graph

$$p(x) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

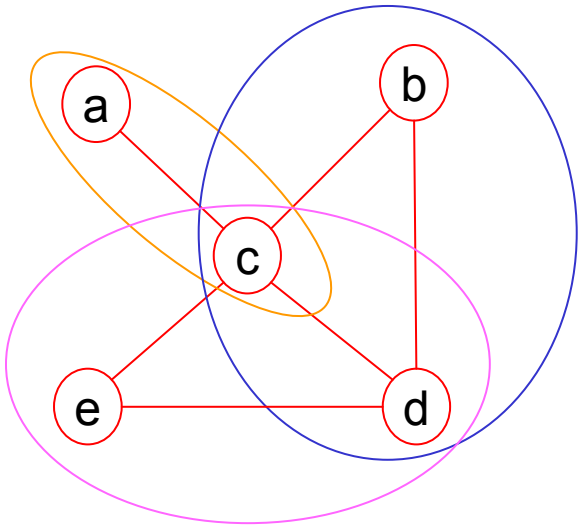
where $\psi_C(x_C)$ are the clique potential, and Z is a normalization constant (sometimes called the partition function)



$$p(x) = \frac{1}{Z} \psi_A(a,c) \psi_B(b,c,d) \psi_C(c,d,e)$$

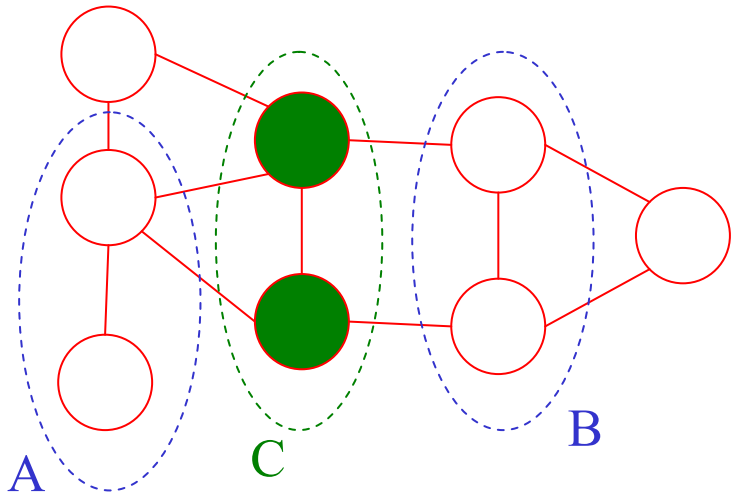
Clique Potentials

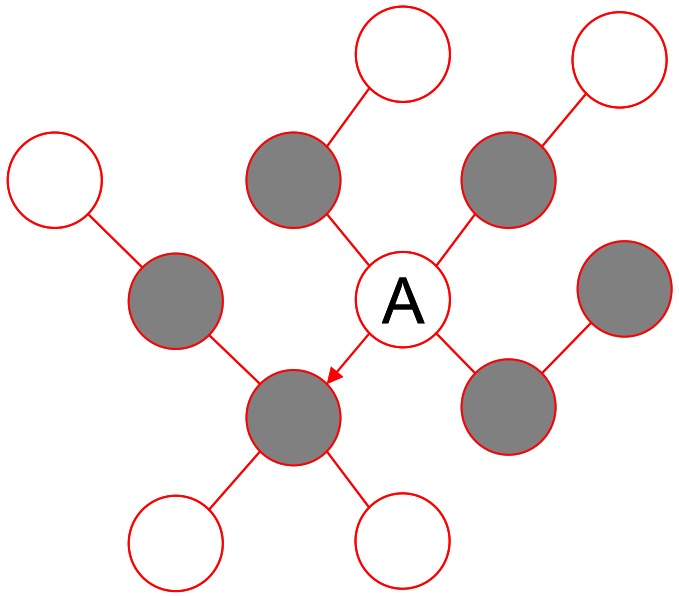
- A clique is a fully connected subgraph
  - By clique we usually mean maximal clique (i.e. not contained within another clique)
  - measures “compatibility” between settings of the variables



Undirected Graphs: Conditional Independence

- $A \perp\!\!\!\perp B \mid C$ simple graph separation can tell us about conditional independencies



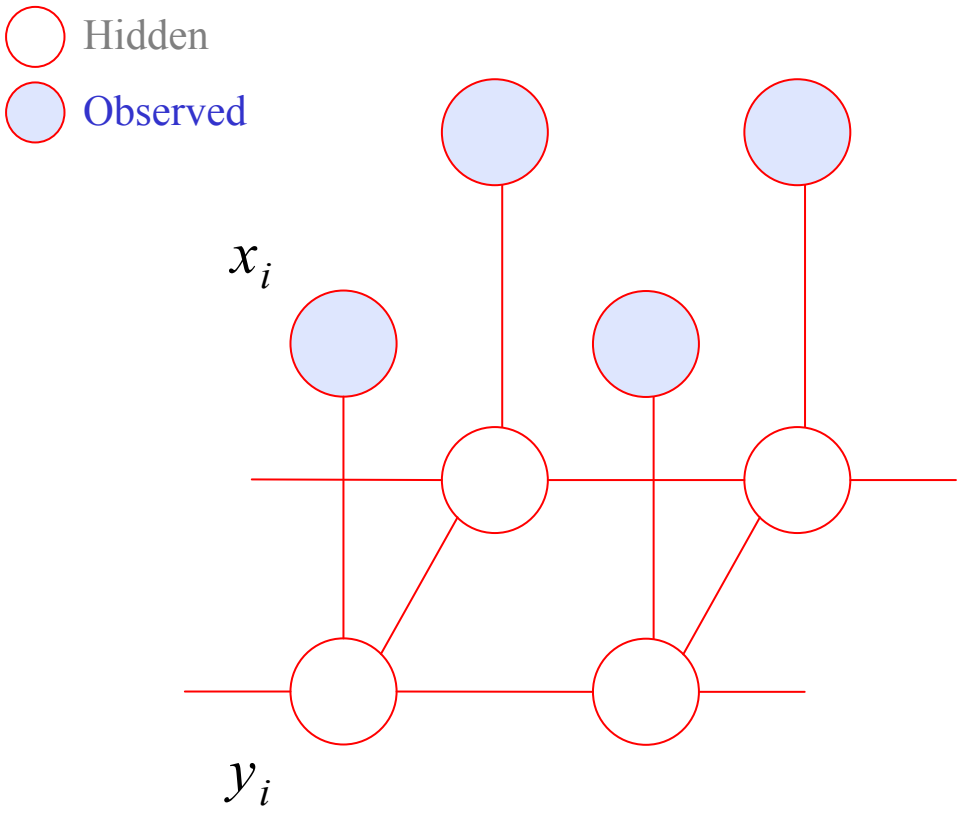


- The Markov blanket of a node A is defined as
  - MB(A)={Neighbors(A)}

Examples of Undirected Graphs

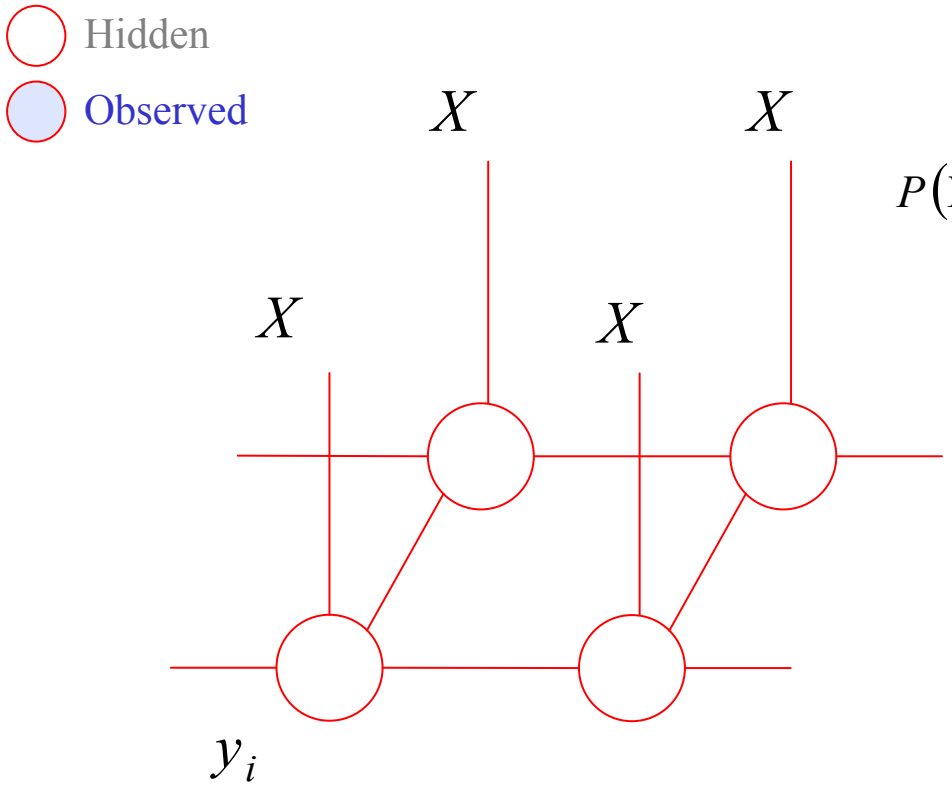
- Markov Random Fields
- Condition Random Fields
- Maximum Entropy Markov Models
- Maximum Entropy
- Boltzmann Machines
- etc,…

Example: Markov Random Field



$$P(X,Y) = \frac{1}{Z}\prod_i \psi_i(x_i, y_i)\prod_{j,k} \psi_{jk}(y_j, y_k)$$

Example: Conditional Random Field



$$P(Y \mid X) = \frac{1}{Z} \prod_i \psi_i(y_i \mid X) \prod_{j,k} \psi_{jk}(y_j, y_k \mid X)$$

Summary of Factorization Properties

- Directed graphs

$$p(x_1, \cdots, x_K) = \prod_{k=1}^{K} p(x_k \mid parent\ (x_k))$$

  - Conditional independence from d-separation test
  - Directed graphs are better at expressing causal generative models
- Undirected graphs

$$p(x) = \frac{1}{Z} \prod_{C} \psi_C(x_C)$$

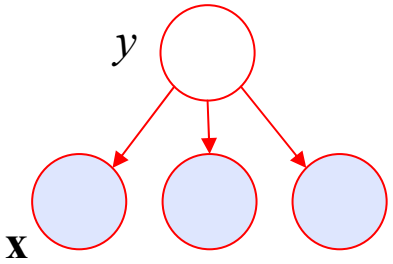
  - Conditional independence from graph separation
  - Undirected graphs are better at representing soft constraints between variables

Applications of Graphical Models

Classification

- Classification is predicting a single class variable y give a vector of feature $\mathbf{x} = (x_1, x_2, \cdots x_K)$
- Naïve Bayes classifier
  - Assume that once the class label is known, all the features are independent
    - based directly on joint probability distribution $p(y, \mathbf{x})$
    - in generative models set of parameters must represent input distribution and conditional well

$$p(y, \mathbf{x}) = p(y) \prod_{i=1}^{K} p(x_i \mid y)$$

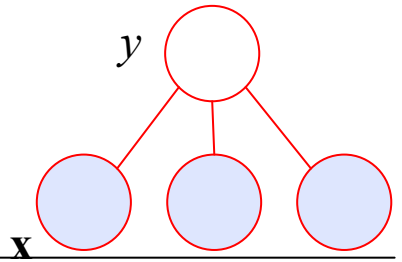


- Logistic regression (maximum entropy classifier)
  - based directly on conditional probability $p(y \mid \mathbf{x})$ ➔need no model $p(\mathbf{x})$
  - in discriminative models are not as strongly tied to their input distribution

class bias weight

$$p(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right\}$$

$$\text{where } Z(\mathbf{x}) = \sum_y \exp\left\{\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right\}$$



* It can be shown that a Gaussian Naïve Bayes (GNB) classifier implies the parametric form of p(y|x) of its discriminative pair logistic regression

Classification (cont.)

Naïve Bayes

$$p(y,\mathbf{x}) = p(y)\prod_{k=1}^{K} p(x_k \mid y)$$

- Consider a GNB based on the following modeling assumptions
  - $P(x_i \mid y = y_k)$ is a Gaussian distribution of the form $N(\mu_{ik}.\sigma_i)$
  - y is Boolean, governed by a Bernoulli distribution with parameter $\theta = P(y=1)$

$$p(y=1 \mid \mathbf{x}) = \frac{p(y=1)p(\mathbf{x} \mid y=1)}{p(y=1)p(\mathbf{x} \mid y=1) + p(y=0)p(\mathbf{x} \mid y=0)}$$
$$= \frac{1}{1 + \frac{p(y=0)p(\mathbf{x} \mid y=0)}{p(y=1)p(\mathbf{x} \mid y=1)}}$$
$$= \frac{1}{1 + \exp\left(\log \frac{p(y=0)p(\mathbf{x} \mid y=0)}{p(y=1)p(\mathbf{x} \mid y=1)}\right)}$$
$$= \frac{1}{1 + \exp\left(\log \frac{1-\theta}{\theta} + \sum_i \log \frac{p(x_i \mid y=0)}{p(x_i \mid y=1)}\right)}$$

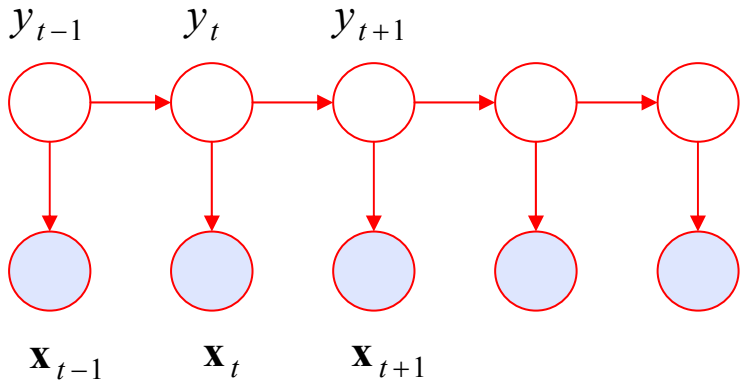
$$\sum_i \log \frac{p(x_i \mid y=0)}{p(x_i \mid y=1)} = \sum_i \log \frac{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(x_i - \mu_{i0})^2}{2\pi\sigma_i^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(x_i - \mu_{i1})^2}{2\pi\sigma_i^2}\right)}$$
$$= \sum_i \log \exp\left(\frac{(x_i - \mu_{i1})^2 - (x_i - \mu_{i0})^2}{2\pi\sigma_i^2}\right)$$
$$= \sum_i \left(\frac{(x_i - \mu_{i1})^2 - (x_i - \mu_{i0})^2}{2\pi\sigma_i^2}\right)$$
$$= \sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} x_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\pi\sigma_i^2}\right)$$

$$p(y=1 \mid \mathbf{x}) = \frac{1}{1 + \exp\left(\log \frac{1-\theta}{\theta} + \sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} x_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\pi\sigma_i^2}\right)\right)}$$
$$= \frac{1}{1 + \exp\left(\lambda_0 + \sum_i \lambda_i x_i\right)}$$

Sequence Models

- Classifier predict only a single class variable, but the true power of graphical models lies in their ability to model many variables that are interdependent
  - e.g. named-entity recognition (NER), part-of-speech tagging (POS)
- Hidden Markov models
  - Relax the independence assumption by arranging the output variables in a linear chain
  - To model the joint distribution $p(\mathbf{Y}, \mathbf{X})$, an HMM makes two assumptions
    - Each state depends only on its immediate predecessor (First order assumption)
    - Each observation variable depends only on the current state (Output-independent assumption)

$$p(\mathbf{Y}, \mathbf{X}) = p(y_0) \prod_{t=1}^{T} p(y_t \mid y_{t-1}) p(\mathbf{x}_t \mid y_t)$$



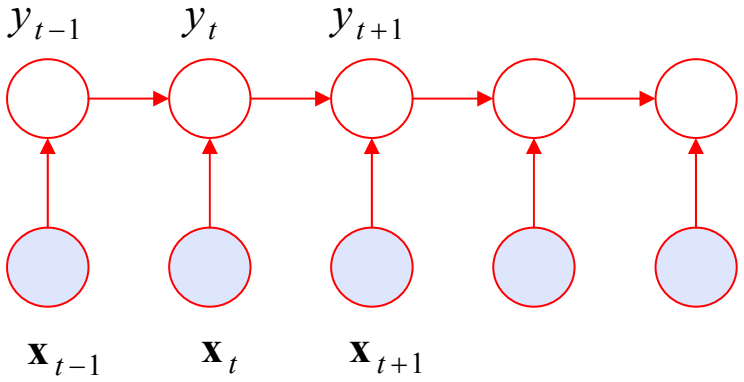
Sequence Models (cont.)

- Maximum Entropy Markov Models (MEMMs)
  - A conditional model that representing the probability of reaching a state given an observation and the previous state

$$p(\mathbf{Y} \mid \mathbf{X}) = p(y_1 \mid x_1) \prod_{t=2}^{T} p(y_t \mid y_{t-1}, x_t)$$

$$p(y_t \mid y_{t-1}, x_t) = \frac{1}{Z} \exp\left(\sum_k \lambda_k f_k (y_{t-1}, y_t, x_t) \right)$$

$$Z = \sum_{y'} \exp\left(\sum_k \lambda_k f_k (y_{t-1}, y'_t, x_t) \right)$$



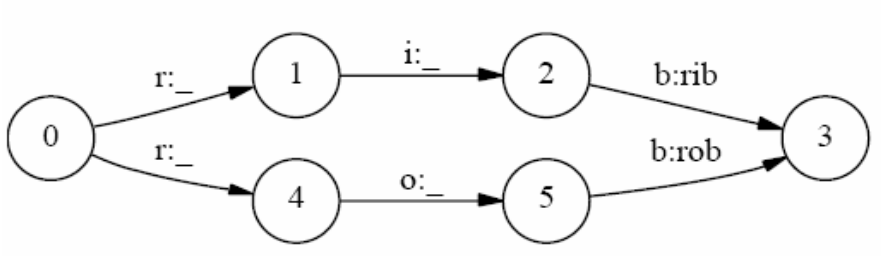
* per-state normalization

  - Per-state normalization will cause all the mass that arrives at a state must be distributed among the possible successor states

Label Bias Problem!!!!!
Potential victims: Discriminative Models

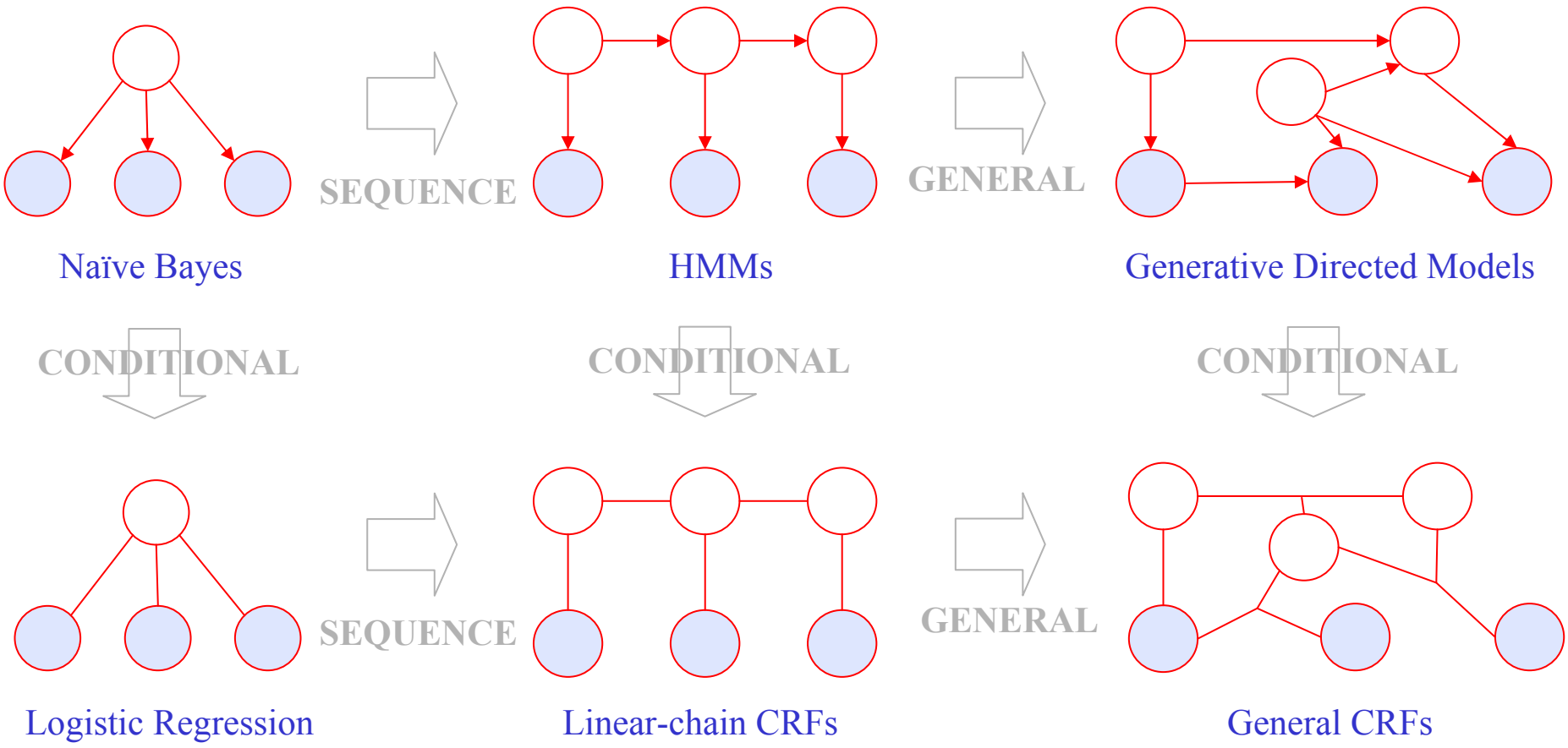
Sequence Models (cont.)

- Label Bias Problem
  - Consider this MEMM



  - P(1 and 2 | ro) = P(2 | 1 and ro)P(1 | ro) = P(2 | 1 and o)P(1 | r)
    P(1 and 2 | ri) = P(2 | 1 and ri)P(1 | ri) = P(2 | 1 and i)P(1 | r)
  - Since P(2 | 1 and x) = 1 for all x, P(1 and 2 | ro) = P(1 and 2 | ri)
    - In the training data, label 2 is the only label value observed after label 1
      Therefore P(2 | 1) = 1, so P(2 | 1 and x) = 1 for all x
  - However, we expect P(1 and 2 | ri) to be greater than P(1 and 2 | ro)

Sequence Models (cont.)



From HMM to CRFs

HMMs

$$p(y,x) = p(y_0)\prod_{t=1}^{T} p(y_t \mid y_{t-1})p(x_t \mid y_t)$$

- We can rewrite the HMM joint distribution $p(y,x)$ as follows

$$p(\mathbf{Y},\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_t \sum_{i,j\in S} \lambda_{ij} 1_{\{y_t=i\}} 1_{\{y_{t-1}=j\}} + \sum_t \sum_{i\in S} \sum_{\mathbf{o}\in O} \mu_{oi} 1_{\{y_t=i\}} 1_{\{\mathbf{x}_t=\mathbf{o}\}}\right)$$

  - Because we do not require the parameter to be log probabilities, we are no longer guaranteed that the distribution sums to 1
    - So we explicitly enforce this by using a normalization constant *Z*
- We can write the above equation more compactly by introducing the concept of feature function

$$p(\mathbf{Y},\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

Feature function for HMMs

$f_k(y_t, y_{t-1}, \mathbf{x}_t) = 1_{\{y_t=i\}} 1_{\{y_{t-1}=i\}}$ state transition

$f_k(y_t, y_{t-1}, \mathbf{x}_t) = 1_{\{y_t=i\}} 1_{\{\mathbf{x}=\mathbf{o}\}}$ state observation

- The last step is to write the conditional distribution $p(\mathbf{Y} \mid \mathbf{X})$

$$p(\mathbf{Y} \mid \mathbf{X}) = \frac{p(\mathbf{Y},\mathbf{X})}{\sum_{\mathbf{Y}'} p(\mathbf{Y}',\mathbf{X})} = \frac{\exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}}{\sum_{y'} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y'_t, y'_{t-1}, \mathbf{x}_t)\right\}}$$

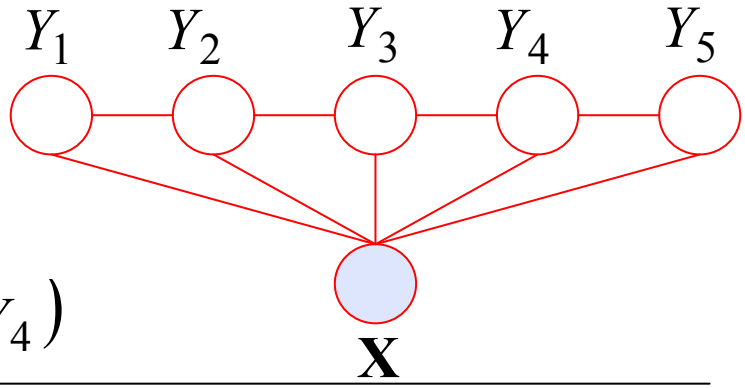
Linear-chain CRFs

More Detail on Conditional Random Fields

Conditional Random Fields

- CRFs have all the advantages of MEMMs without label bias problem
  - MEMM uses per-state exponential model for the conditional probabilities of next states given the current state
  - CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence
- Let $G(V, E)$ be a graph such that $\mathbf{Y} = (Y_v)_{v \in V}$, so that $\mathbf{Y}$ is indexed by the vertices of G. Then $(\mathbf{X}, \mathbf{Y})$ is a conditional random field in case, when conditioned on $\mathbf{X}$, the random variables $\mathbf{Y}_v$ obey the Markovian property

$$p(\mathbf{Y}_v \mid \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v \mid \mathbf{X}, neighbor\ (\mathbf{Y}_v))$$



$$p(Y_3 \mid \mathbf{X}, \text{all other } Y) = p(Y_3 \mid \mathbf{X}, Y_2, Y_4)$$

Linear-Chain Conditional Random Fields

- Definition

  Let Y, X be the random vectors, $\Lambda = \{\lambda_k\} \in R^K$ be a parameter vector, and $\{f_k(y, y', \mathbf{x}_t)\}_{k=1}^K$ be a set of real-valued feature functions. Then a linear-chain conditional random field is a distribution $p(\mathbf{y} \mid \mathbf{x})$ that takes the form

$$p(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right) \quad \text{or} \quad p(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\left(\boldsymbol{\Lambda}^T \mathbf{F}\right)$$

  Where $Z(\mathbf{X})$ is an instance-specific normalization function

$$Z(\mathbf{X}) = \sum_{\mathbf{Y}} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

  sum over all possible state sequences
  an exponentially large number of terms

Fortunately, forward-backward indeed helps us to calculate this term

Forward and Backward Algorithms

- Suppose that we are interested in tagging a sequence only partially, say till the position *i*
  - Denote the un-normalized probability of a partial labeling ending at position *i* with fixed label *y* by $\alpha(y,i)$
  - Denote the un-normalized probability of a partial segmentation starting at position *i+1* assuming a label *y* at position *i* by $\beta(y,i)$

    α and β can be computed via the following recurrences

$$\alpha(y,i) = \sum_{y'} a(y',i-1) \times \exp\left(\boldsymbol{\Lambda}^T \mathbf{f}(y,y',\mathbf{x}_i)\right)$$

$$\beta(y,i) = \sum_{y'} \beta(y',i+1) \times \exp\left(\boldsymbol{\Lambda}^T \mathbf{f}(y,y',\mathbf{x}_{i+1})\right)$$

  - We can now write the marginal and partition function in term of these

$$P(Y_i = y \mid \mathbf{X}) = \alpha(y,i)\beta(y,i)/Z(\mathbf{X})$$

$$P(Y_i = y, Y_{i+1} = y' \mid \mathbf{X}) = \alpha(y,i)\exp\left(\boldsymbol{\Lambda}\mathbf{f}(y',y,\mathbf{x}_{i+1})\right)\beta(y',i+1)/Z(\mathbf{X})$$

$$Z(\mathbf{X}) = \sum_{y} \alpha(y,|\mathbf{X}|) = \sum_{y} \beta(y,1)$$

Inference in linear CRFs using the Viterbi Algorithm

- Given the parameter vector Λ, the best labeling for a sequence can be found exactly using the Viterbi algorithm
  - For each tuple of the form (i, y), the Viterbi algorithm maintains the un-normalized probability of the best labeling ending at position *i* with the label *y*
  - The recurrence is

$$V(i,y) = \begin{cases} \max_{y'} \left(V(i-1,y') \times \exp\left(\Lambda^T \mathbf{f}(y, y', \mathbf{x}_i) \right) \right) & (\text{i} > 0) \\ [[\, y = start \,]] & (\text{I} = 0) \end{cases}$$

- The normalized probability of the best labeling is given by

$$\frac{\max_y V(n,y)}{Z(\mathbf{X})}$$

Training (Parameter Estimation)

- The various methods used to train CRFs differ mainly in the objective function they try to optimize
  - Penalized log-likelihood criteria
  - Voted perceptron
  - Pseudo log-likelihood
  - Margin maximization
  - Gradient tree boosting
  - Logarithmic pooling
  - and so on …

Penalized log-likelihood criteria

- The conditional log-likelihood of a set of training instances $(\mathbf{X}^k, \mathbf{Y}^k)$ using parameters $\mathbf{\Lambda}$ is given by

$$L_{\mathbf{\Lambda}} = \sum_k \mathbf{\Lambda}^T \mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \log Z_{\mathbf{\Lambda}}(\mathbf{X}^k)$$

The gradient of the log-likelihood is given by

$$\begin{aligned}
\nabla L_{\mathbf{\Lambda}} &= \sum_k \left(\mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \frac{\sum_{\mathbf{Y}'} \mathbf{F}(\mathbf{Y}', \mathbf{X}^k) \exp\left(\mathbf{\Lambda}^T \mathbf{F}(\mathbf{Y}', \mathbf{X}^k)\right)}{Z_{\mathbf{\Lambda}}(\mathbf{X}^k)} \right) \\
&= \sum_k \left(\mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \sum_{\mathbf{Y}'} \mathbf{F}(\mathbf{Y}', \mathbf{X}^k) P(\mathbf{Y}' | \mathbf{X}^k) \right) \\
&= \sum_k \left(\mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)} \left[\mathbf{F}(\mathbf{Y}, \mathbf{X}^k) \right] \right)
\end{aligned}$$

In order to avoid overfitting problem, we impose a penalty on it

$$L_{\mathbf{\Lambda}} = \sum_k \left(\mathbf{\Lambda}^T \mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \log Z_{\mathbf{\Lambda}}(\mathbf{X}^k) \right) - \frac{\|\mathbf{\Lambda}\|^2}{2\sigma^2}$$ Euclidean norm

and the gradient is given by

$$\nabla L_{\mathbf{\Lambda}} = \sum_k \left(\mathbf{F}(\mathbf{Y}^k, \mathbf{X}^k) - \mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)} \left[\mathbf{F}(\mathbf{Y}, \mathbf{X}^k) \right] \right) - \frac{\mathbf{\Lambda}}{\sigma^2}$$

Penalized log-likelihood criteria (cont.)

The tricky term in the gradient is the expectation $\mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)}[\mathbf{F}(\mathbf{Y},\mathbf{X}^k)]$ those computation requires the enumeration of all the possible $\mathbf{Y}$ sequence

Let us look at the *j th* entry in this vector, viz. $\mathbf{F}_j(\mathbf{Y},\mathbf{X}^k)$ and $\mathbf{F}_j(\mathbf{Y},\mathbf{X}^k)$ is equal to $\sum_i f_j(y_i, y_{i-1}, \mathbf{X}_i^k)$. Therefore, we can rewrite $\mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)}[\mathbf{F}(\mathbf{Y},\mathbf{X}^k)]$ as

$$\begin{aligned}
\mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)}[\mathbf{F}_j(\mathbf{Y},\mathbf{X}^k)] &= \mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)}\left[\sum_i f_j(y_i, y_{i-1}, \mathbf{X}_i^k)\right] \\
&= \sum_i \mathrm{E}_{P(\mathbf{Y}|\mathbf{X}^k)}[f_j(y_i, y_{i-1}, \mathbf{X}_i^k)] \\
&= \sum_i \sum_{y',y} \alpha(i-1, y') f_j(y, y', \mathbf{X}_i^k) \exp\left(\boldsymbol{\Lambda}^T \mathbf{f}(y, y', \mathbf{X}_i^k)\right) \beta(i, y) \\
&= \sum_i \alpha_{i-1}^T Q_i \beta_i
\end{aligned}$$

- After obtained the gradient, various iterative methods can be used to maximized the log-likelihood
  - Improved Iterative Scaling (IIS), Generalized Iterative Scaling (GIS), Limited Memory Quasi Newton method (L-BFGS)

Voted Perceptron Method

- Perceptron uses an approximation of the gradient of the unregularized log-likelihood function $\nabla L_{\Lambda} = \sum_{k} \left(\mathbf{F}\left(\mathbf{Y}^{k}, \mathbf{X}^{k}\right) - \mathrm{E}_{P\left(\mathbf{Y}|\mathbf{X}^{k}\right)} \left[\mathbf{F}\left(\mathbf{Y}, \mathbf{X}^{k}\right) \right] \right)$
  - It considers one misclassified instance at a time, along with its contribution to the gradient $\left(\mathbf{F}\left(\mathbf{Y}^{k}, \mathbf{X}^{k}\right) - \mathrm{E}_{P\left(\mathbf{Y}|\mathbf{X}^{k}\right)} \left[\mathbf{F}\left(\mathbf{Y}, \mathbf{X}^{k}\right) \right] \right)$
  - The feature expectation is further approximated by a point estimate of the feature vector at the best possible labeling

$$\nabla L_{\Lambda} \approx \mathbf{F}\left(\mathbf{Y}^{k}, \mathbf{X}^{k}\right) - \mathbf{F}\left(\mathbf{Y}^{*k}, \mathbf{X}^{k}\right) \quad \left(\mathbf{Y}^{*k} = \arg\max_{\mathbf{Y}} \Lambda^{T} \mathbf{F}\left(\mathbf{Y}, \mathbf{X}^{k}\right) \right)$$

MAP-hypothesis based classifier

Using this approximate gradient, the following first order update rule can be used for maximization

$$\Lambda_{t+1} = \Lambda_{t} + \mathbf{F}\left(\mathbf{Y}^{k}, \mathbf{X}^{k}\right) - \mathbf{F}\left(\mathbf{Y}^{*k}, \mathbf{X}^{k}\right)$$

This update step is applied once for each misclassified instance in the training set. Or we can collect all the update in each pass and take their unweighted average to update the parameter

Pseudo log-likelihood

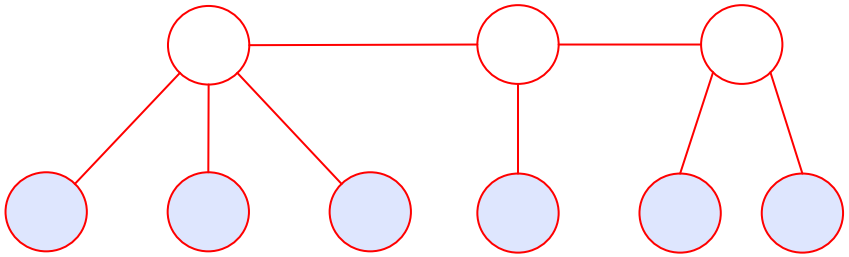
- In many scenarios, we are willing to assign different error values to different labeling
  - It makes senses to maximize the marginal distributions $P\left(y_t^k \mid \mathbf{X}^k\right)$instead of $P\left(\mathbf{Y}^k \mid \mathbf{X}^k\right)$
  - This objective is called the pseudo-likelihood and for the case of linear CRFs, it is given by

$$L_{\Lambda} = \sum_k \sum_{t=1}^{T} \log P\left(y_t^k \mid \mathbf{X}^k, \Lambda\right)$$

$$= \sum_k \sum_{t=1}^{T} \log \sum_{\mathbf{y}: y_t = y_t^k} \frac{\exp\left(\Lambda^T \mathbf{F}\left(\mathbf{y}, \mathbf{X}^k\right)\right)}{Z_{\Lambda}\left(\mathbf{X}^k\right)}$$

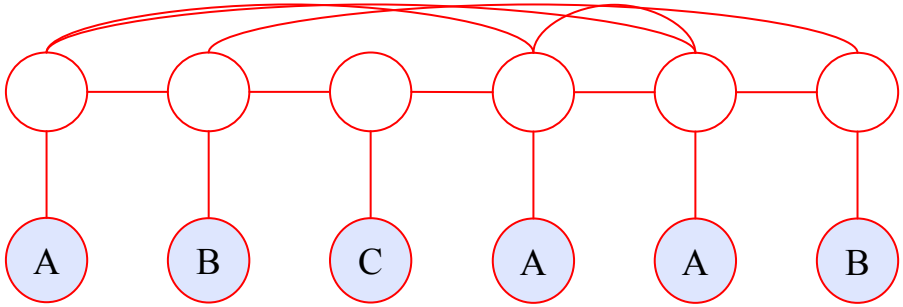
Other types of CRFs

- Semi-Markov CRFs
  - It is still in the realm of first-order Markovian dependence, but the different is the label depend only on segment feature and the label of previous segment
    - Instead of assigning labels to each position, assign labels to segments



Semi-Markov CRFs

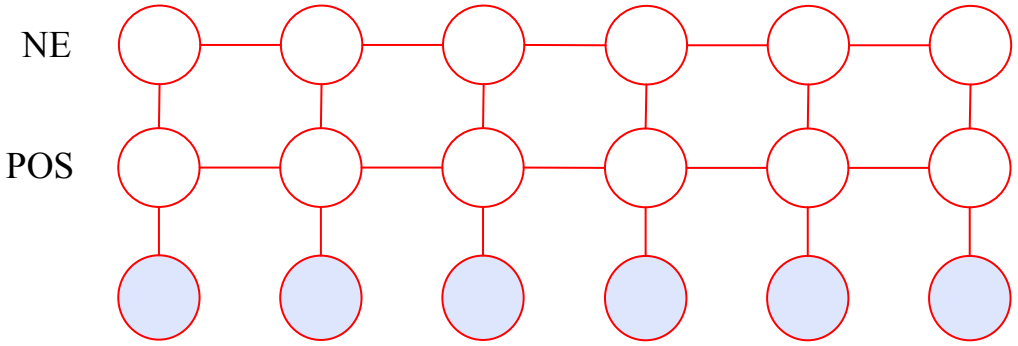
- Skip-Chain CRFs
  - A conditional model that collectively segments a document into mentions and classifies the mentions by entity type



Skip-chain CRFs

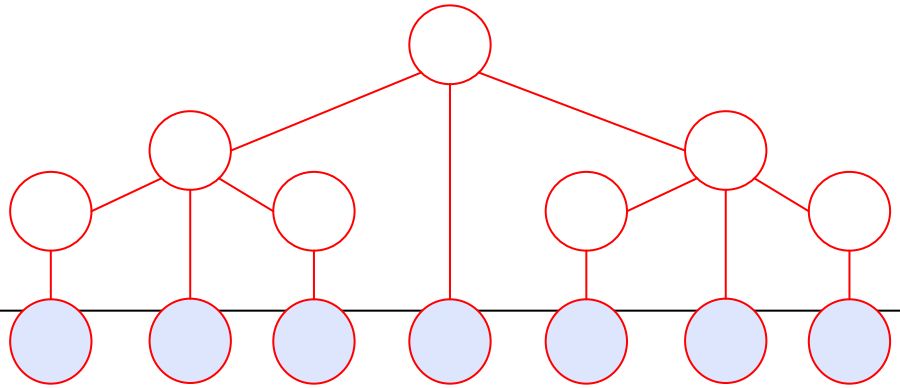
Other types of CRFs (cont.)

- Factorial CRFs
  - Several synchronized inter-dependent tasks
    - Cascading propagates errors



Factorial CRFs

- Tree CRFs
  - The dependencies are organized as a tree structure



Tree CRFs

Conclusions

- Conditional Random Fields offer a unique combination of properties
  - discriminatively trained models for sequence segmentation and labeling
  - combination of arbitrary and overlapping observation features from both the past and future
  - efficient training and decoding based on dynamic programming for a simple chain graph
  - parameter estimation guaranteed to find the global optimum
- Possible Future work?
  - Efficient training approach ??
  - Efficient Feature Induction ??
  - Constrained Inferencing ??
  - Different topology ??

Reference

- Lafferty, J., McCallum, A., Pereira, F., “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 282–289.
- Rahul Gupta, “Conditional Random Fields,” Dept. of Computer Science and Engg., IIT Bombay, India. Document available from http://www.it.iitb.ac.in/~grahul/main.pdf
- Sutton, C., McCallum, A., “An Introduction to Conditional Random Fields for Relational Learning,” Introduction to Statistical Relational Learning, MIT Press. 2007. Document available from http://www.cs.berkeley.edu/~casutton/publications/crf-tutorial.pdf
- Mitchell, T. M., “Machine Learning,” McGraw Hill, 1997. Document available from http://www.cs.cmu.edu/~tom/mlbook.html
- Bishop, C. M., “Pattern Recognition and Machine Learning,” Springer, 2006
- Bishop, C. M., “Graphical Models and Variational Methods,” video lecture, Machine Learning Summer School 2004. Available from http://videolectures.net/mlss04_bishop_gmvm/
- Ghahramani, Z., “Graphical models,” video lecture, EPSRC Winter School in Mathematics for Data Modelling, 2008. Available from http://videolectures.net/epsrcws08_ghahramani_gm/
- Roweis, S., “Machine Learning, Probability and Graphical Models,” video lecture, Machine Learning Summer School 2006, Available from http://videolectures.net/mlss06tw_roweis_mlpgm/