Probabilistic Graphical Models

David Sontag

New York University

Lecture 3, February 14, 2013

Undirected graphical models

Reminder of lecture 2

- An alternative representation for joint distributions is as an **undirected graphical model** (also known as **Markov random fields**)
- As in BNs, we have one node for each random variable
- Rather than CPDs, we specify (non-negative) **potential functions** over sets of variables associated with cliques C of the graph,

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \phi_c(\mathbf{x}_c)$$

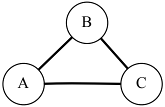
Z is the **partition function** and normalizes the distribution:

$$Z = \sum_{\hat{x}_1, \ldots, \hat{x}_n} \prod_{c \in C} \phi_c(\hat{\mathbf{x}}_c)$$

Undirected graphical models

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \phi_c(\mathbf{x}_c), \qquad Z = \sum_{\hat{x}_1, \ldots, \hat{x}_n} \prod_{c \in C} \phi_c(\hat{\mathbf{x}}_c)$$

Simple example (potential function on each edge encourages the variables to take the same value):



$\phi_{A,B}(a,b) =$

A \ B	0	1
0	10	1
1	1	10

$\phi_{B,C}(b,c) =$

B \ C	0	1
0	10	1
1	1	10

$\phi_{A,C}(a,c) =$

A \ C	0	1
0	10	1
1	1	10

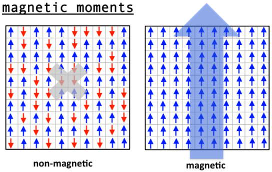
$$p(a, b, c) = \frac{1}{Z} \phi_{A,B}(a, b) \cdot \phi_{B,C}(b, c) \cdot \phi_{A,C}(a, c),$$

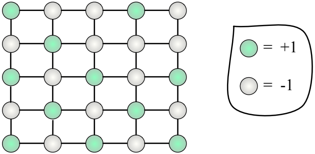
where

$$Z = \sum_{\hat{a}, \hat{b}, \hat{c} \in \{0,1\}^3} \phi_{A,B}(\hat{a}, \hat{b}) \cdot \phi_{B,C}(\hat{b}, \hat{c}) \cdot \phi_{A,C}(\hat{a}, \hat{c}) = 2 \cdot 1000 + 6 \cdot 10 = 2060.$$

Example: Ising model

- Invented by the physicist Wilhelm Lenz (1920), who gave it as a problem to his student Ernst Ising
- Mathematical model of ferromagnetism in statistical mechanics
- The spin of an atom is biased by the spins of atoms nearby on the material:

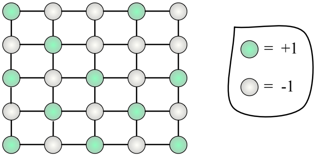




- Each atom $X_i \in \{-1, +1\}$, whose value is the direction of the atom spin
- If a spin at position i is $+1$, what is the probability that the spin at position j is also $+1$?
- Are there phase transitions where spins go from "disorder" to "order"?

Example: Ising model

- Each atom $X_i \in \{-1, +1\}$, whose value is the direction of the atom spin
- The spin of an atom is biased by the spins of atoms nearby on the material:



$$p(x_1, \cdots, x_n) = \frac{1}{Z} \exp\left(\sum_{i<j} w_{i,j} x_i x_j - \sum_i u_i x_i \right)$$

- When $w_{i,j} > 0$, nearby atoms encouraged to have the same spin (called **ferromagnetic**), whereas $w_{i,j} < 0$ encourages $X_i \neq X_j$
- Node potentials $\exp(-u_i x_i)$ encode the bias of the individual atoms
- Scaling the parameters makes the distribution more or less spiky

Today's lecture

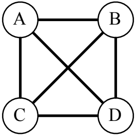
- Markov random fields
  1. Factor graphs
  2. Bayesian networks $\Rightarrow$ Markov random fields (*moralization*)
  3. Hammersley-Clifford theorem (conditional independence $\Rightarrow$ joint distribution factorization)
- Conditional models
  3. Discriminative versus generative classifiers
  4. Conditional random fields

Higher-order potentials

- The examples so far have all been **pairwise MRFs**, involving only node potentials $\phi_i(X_i)$ and pairwise potentials $\phi_{i,j}(X_i, X_j)$
- Often we need **higher-order** potentials, e.g.

$$\phi(x, y, z) = 1[x + y + z \geq 1],$$

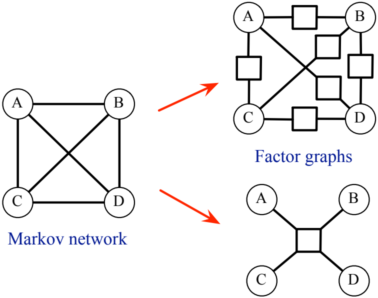
  where X, Y, Z are binary, enforcing that at least one of the variables takes the value 1
- Although Markov networks are useful for understanding independencies, they hide much of the distribution's structure:



Does this have pairwise potentials, or one potential for all 4 variables?

Factor graphs

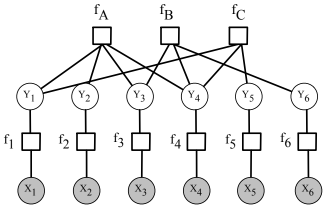
- G does not reveal the structure of the distribution: maximum cliques vs. subsets of them
- A **factor graph** is a bipartite undirected graph with variable nodes and factor nodes. Edges are only between the variable nodes and the factor nodes
- Each factor node is associated with a single potential, whose scope is the set of variables that are neighbors in the factor graph



- The distribution is same as the MRF – this is just a different data structure

Example: Low-density parity-check codes

- Error correcting codes for transmitting a message over a noisy channel (invented by Galleger in the 1960's, then re-discovered in 1996)



- Each of the top row factors enforce that its variables have even parity:

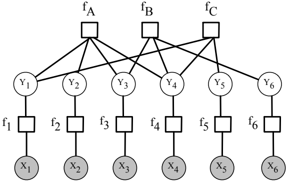
$$f_A(Y_1, Y_2, Y_3, Y_4) = 1 \text{ if } Y_1 \otimes Y_2 \otimes Y_3 \otimes Y_4 = 0, \text{ and } 0 \text{ otherwise}$$

- Thus, the only assignments **Y** with non-zero probability are the following (called **codewords**): *3 bits encoded using 6 bits*

  000000, 011001, 110010, 101011, 111100, 100101, 001110, 010111

- $f_i(Y_i, X_i) = p(X_i \mid Y_i)$, the likelihood of a bit flip according to noise model

Example: Low-density parity-check codes



- The *decoding* problem for LDPCs is to find

$$\mathrm{argmax}_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x})$$

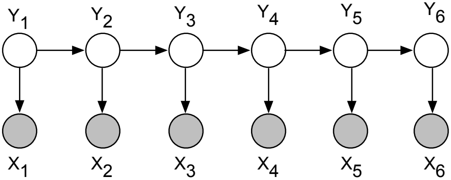
  This is called the **maximum a posteriori** (MAP) assignment
- Since Z and $p(\mathbf{x})$ are constants with respect to the choice of $\mathbf{y}$, can equivalently solve (taking the log of $p(\mathbf{y}, \mathbf{x})$):

$$\mathrm{argmax}_{\mathbf{y}} \sum_{c \in C} \theta_c(\mathbf{x}_c),$$

  where $\theta_c(\mathbf{x}_c) = \log \phi_c(\mathbf{x}_c)$
- This is a discrete optimization problem!

Converting BNs to Markov networks

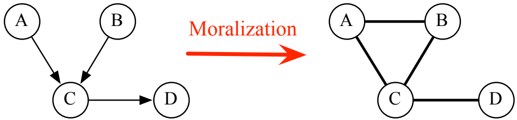
What is the equivalent Markov network for a hidden Markov model?



Many inference algorithms are more conveniently given for undirected models – this shows how they can be applied to Bayesian networks

Moralization of Bayesian networks

- Procedure for converting a Bayesian network into a Markov network
- The **moral graph** $\mathcal{M}[G]$ of a BN $G = (V, E)$ is an undirected graph over V that contains an undirected edge between X_i and X_j if
  1. there is a directed edge between them (in either direction)
  2. X_i and X_j are both parents of the same node



(term historically arose from the idea of "marrying the parents" of the node)

- The addition of the moralizing edges leads to the loss of some independence information, e.g., $A \rightarrow C \leftarrow B$, where $A \perp B$ is lost

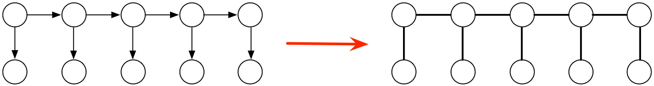
Converting BNs to Markov networks

1. Moralize the directed graph to obtain the undirected graphical model:

2. Introduce one potential function for each CPD:

$$\phi_i(x_i, \mathbf{x}_{pa(i)}) = p(x_i \mid \mathbf{x}_{pa(i)})$$

- So, converting a hidden Markov model to a Markov network is simple:



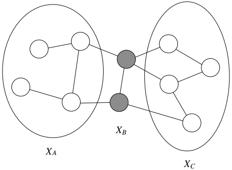
- For variables having > 1 parent, factor graph notation is useful

Factorization implies conditional independencies

- $p(\mathbf{x})$ is a *Gibbs distribution* over G if it can be written as

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \phi_c(\mathbf{x}_c),$$

  where the variables in each potential $c \in C$ form a clique in G
- Recall that conditional independence is given by graph separation:



- Theorem (**soundness of separation**): If $p(\mathbf{x})$ is a Gibbs distribution for G, then G is an I-map for $p(\mathbf{x})$, i.e. $I(G) \subseteq I(p)$
  Proof: Suppose $\mathbf{B}$ separates $\mathbf{A}$ from $\mathbf{C}$. Then we can write

$$p(\mathbf{X_A}, \mathbf{X_B}, \mathbf{X_C}) = \frac{1}{Z} f(\mathbf{X_A}, \mathbf{X_B}) g(\mathbf{X_B}, \mathbf{X_C}).$$

Conditional independencies implies factorization

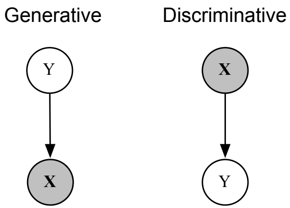
- Theorem (**soundness of separation**): If $p(\mathbf{x})$ is a Gibbs distribution for G, then G is an I-map for $p(\mathbf{x})$, i.e. $I(G) \subseteq I(p)$
- What about the converse? We need one more assumption:
- A distribution is **positive** if $p(\mathbf{x}) > 0$ for all $\mathbf{x}$
- Theorem (**Hammersley-Clifford**, 1971): If $p(\mathbf{x})$ is a positive distribution and G is an I-map for $p(\mathbf{x})$, then $p(\mathbf{x})$ is a Gibbs distribution that factorizes over G
- Proof is in book (as is counter-example for when $p(\mathbf{x})$ is not positive)
- This is important for **learning**:
  - Prior knowledge is often in the form of conditional independencies (i.e., a graph structure G)
  - Hammersley-Clifford tells us that it suffices to search over Gibbs distributions for G – allows us to *parameterize* the distribution

Today's lecture

- Markov random fields
  1. Factor graphs
  2. Bayesian networks $\Rightarrow$ Markov random fields (*moralization*)
  3. Hammersley-Clifford theorem (conditional independence $\Rightarrow$ joint distribution factorization)
- Conditional models
  3. Discriminative versus generative classifiers
  4. Conditional random fields

Discriminative versus generative classifiers

- There is often significant flexibility in choosing the structure and parameterization of a graphical model

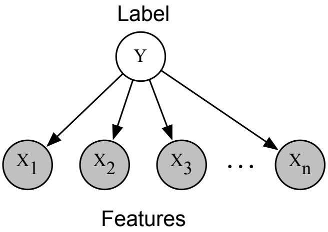


It is important to understand the trade-offs

- In the next few slides, we will study this question in the context of e-mail classification

From lecture 1... naive Bayes for single label prediction

- Classify e-mails as spam ($Y = 1$) or not spam ($Y = 0$)
  - Let $1 : n$ index the words in our vocabulary (e.g., English)
  - $X_i = 1$ if word i appears in an e-mail, and 0 otherwise
  - E-mails are drawn according to some distribution $p(Y, X_1, \ldots, X_n)$
- Words are conditionally independent given Y:

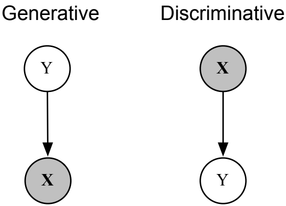


- Prediction given by:

$$p(Y = 1 \mid x_1, \ldots x_n) = \frac{p(Y = 1) \prod_{i=1}^{n} p(x_i \mid Y = 1)}{\sum_{y=\{0,1\}} p(Y = y) \prod_{i=1}^{n} p(x_i \mid Y = y)}$$

Discriminative versus generative models

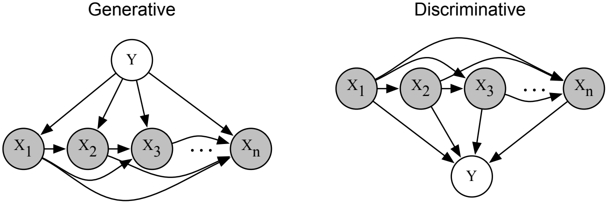
- Recall that these are **equivalent** models of $p(Y, \mathbf{X})$:



- However, suppose all we need for prediction is $p(Y \mid \mathbf{X})$
- In the left model, we need to estimate *both* $p(Y)$ and $p(\mathbf{X} \mid Y)$
- In the right model, it suffices to estimate just the **conditional distribution** $p(Y \mid \mathbf{X})$
  - We never need to estimate $p(\mathbf{X})$!
  - Would need $p(\mathbf{X})$ if $\mathbf{X}$ is only partially observed
  - Called a **discriminative** model because it is only useful for discriminating Y's label

Discriminative versus generative models

- Let's go a bit deeper to understand what are the trade-offs inherent in each approach
- Since $\mathbf{X}$ is a random vector, for $Y \rightarrow \mathbf{X}$ to be equivalent to $\mathbf{X} \rightarrow Y$, we must have:



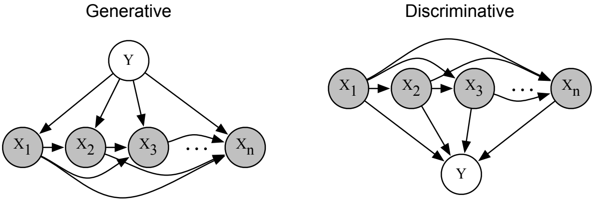
We must make the following choices:

1. In the generative model, how do we parameterize $p(X_i \mid \mathbf{X}_{pa(i)}, Y)$?
2. In the discriminative model, how do we parameterize $p(Y \mid \mathbf{X})$?

Discriminative versus generative models

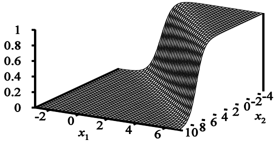
We must make the following choices:

1. In the generative model, how do we parameterize $p(X_i \mid \mathbf{X}_{pa(i)}, Y)$?
2. In the discriminative model, how do we parameterize $p(Y \mid \mathbf{X})$?



1. For the generative model, assume that $X_i \perp \mathbf{X}_{-i} \mid Y$ (**naive Bayes**)
2. For the discriminative model, assume that

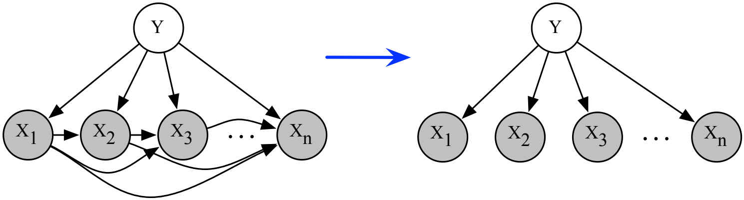
$$p(Y = 1 \mid \mathbf{x}; \alpha) = \frac{1}{1 + e^{-\alpha_0 - \sum_{i=1}^{n} \alpha_i x_i}}$$



This is called **logistic regression**. *(To simplify the story, we assume $X_i \in \{0, 1\}$)*

Naive Bayes

1. For the generative model, assume that $X_i \perp \mathbf{X}_{-i} \mid Y$ (**naive Bayes**)



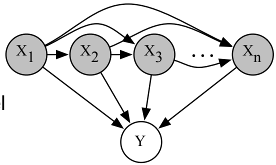
Logistic regression

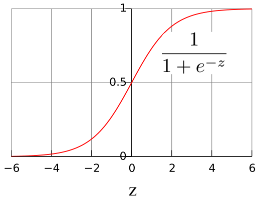
2. For the discriminative model, assume that

$$p(Y = 1 \mid \mathbf{x}; \alpha) = \frac{e^{\alpha_0 + \sum_{i=1}^{n} \alpha_i x_i}}{1 + e^{\alpha_0 + \sum_{i=1}^{n} \alpha_i x_i}} = \frac{1}{1 + e^{-\alpha_0 - \sum_{i=1}^{n} \alpha_i x_i}}$$

Let $z(\alpha, \mathbf{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$.Then, $p(Y = 1 \mid \mathbf{x}; \alpha) = f(z(\alpha, \mathbf{x}))$, where $f(z) = 1/(1 + e^{-z})$ is called the **logistic function**:

Same graphical model





Discriminative versus generative models

1. For the generative model, assume that $X_i \perp \mathbf{X}_{-i} \mid Y$ (**naive Bayes**)
2. For the discriminative model, assume that

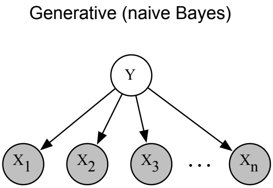
$$p(Y = 1 \mid \mathbf{x}; \alpha) = \frac{e^{\alpha_0 + \sum_{i=1}^{n} \alpha_i x_i}}{1 + e^{\alpha_0 + \sum_{i=1}^{n} \alpha_i x_i}} = \frac{1}{1 + e^{-\alpha_0 - \sum_{i=1}^{n} \alpha_i x_i}}$$

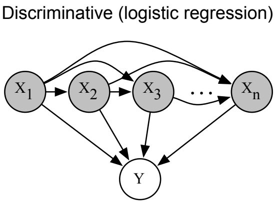
- In problem set 1, you showed **assumption** 1 $\Rightarrow$ **assumption** 2
- Thus, every conditional distribution that can be represented using naive Bayes can *also* be represented using the logistic model

- What can we conclude from this?

With a large amount of training data, logistic regression will perform at least as well as naive Bayes!

Discriminative models are powerful

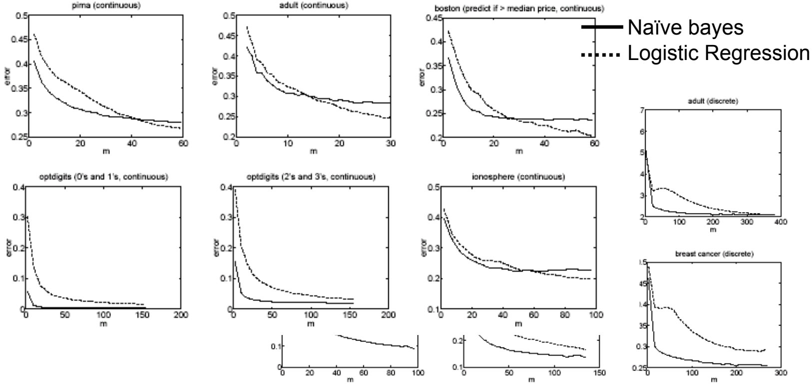




- Logistic model does *not* assume $X_i \perp \mathbf{X}_{-i} \mid Y$, unlike naive Bayes
- This can make a big difference in many applications
- For example, in spam classification, let $X_1 = 1[\text{“bank” in e-mail}]$ and $X_2 = 1[\text{“account” in e-mail}]$
- Regardless of whether spam, these always appear together, i.e. $X_1 = X_2$
- Learning in naive Bayes results in $p(X_1 \mid Y) = p(X_2 \mid Y)$. Thus, naive Bayes **double counts the evidence**
- Learning with logistic regression sets $\alpha_i = 0$ for one of the words, in effect ignoring it (there are other equivalent solutions)

Generative models are still very useful

1. Using a conditional model is only possible when $\mathbf{X}$ is always observed
   - When some X_i variables are unobserved, the generative model allows us to compute $p(Y \mid \mathbf{X}_e)$ by marginalizing over the unseen variables
2. Estimating the generative model using maximum likelihood is more **efficient** (statistically) than discriminative training [Ng & Jordan, 2002]
   - Amount of training data needed to get close to infinite data solution
   - Naive Bayes needs $O(\log n)$ samples
   - Logistic regression needs $O(n)$ samples
   - **Naive Bayes converges more quickly to its (perhaps less helpful) asymptotic estimates**



Some experiments from UCI data sets

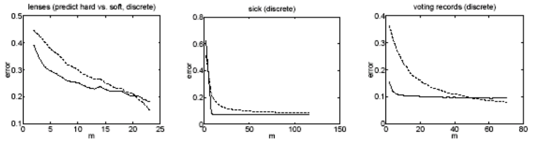


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

Conditional random fields (CRFs)

- **Conditional random fields** are undirected graphical models of conditional distributions $p(\mathbf{Y} \mid \mathbf{X})$
  - $\mathbf{Y}$ is a set of **target variables**
  - $\mathbf{X}$ is a set of **observed variables**
- We typically show the graphical model using just the $\mathbf{Y}$ variables
- Potentials are a function of $\mathbf{X}$ and $\mathbf{Y}$

Formal definition

- A CRF is a Markov network on variables $\mathbf{X} \cup \mathbf{Y}$, which specifies the conditional distribution

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \phi_c(\mathbf{x}_c, \mathbf{y}_c)$$

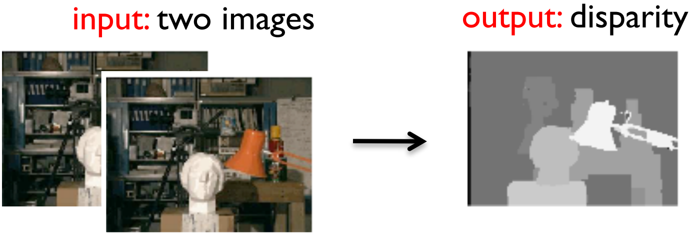
with partition function

$$Z(\mathbf{x}) = \sum_{\hat{\mathbf{y}}} \prod_{c \in C} \phi_c(\mathbf{x}_c, \hat{\mathbf{y}}_c).$$

- As before, two variables in the graph are connected with an undirected edge if they appear together in the scope of some factor
- The only difference with a standard Markov network is the normalization term – before marginalized over $\mathbf{X}$ and $\mathbf{Y}$, now only over $\mathbf{Y}$

CRFs in computer vision

- Undirected graphical models very popular in applications such as computer vision: segmentation, stereo, de-noising
- Grids are particularly popular, e.g., pixels in an image with 4-connectivity



- Not encoding $p(\mathbf{X})$ is the main strength of this technique, e.g., if $\mathbf{X}$ is the image, then we would need to encode the distribution of natural images!
- Can encode a rich set of features, without worrying about their distribution

Parameterization of CRFs

- Factors may depend on a large number of variables
- We typically parameterize each factor as a log-linear function,

$$\phi_c(\mathbf{x}_c, \mathbf{y}_c) = \exp\{\mathbf{w} \cdot \mathbf{f}_c(\mathbf{x}_c, \mathbf{y}_c)\}$$

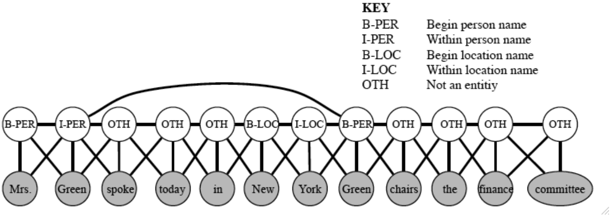
- $\mathbf{f}_c(\mathbf{x}_c, \mathbf{y}_c)$ is a feature vector
- $\mathbf{w}$ is a weight vector which is typically learned – we will discuss this extensively in later lectures

NLP example: named-entity recognition

- Given a sentence, determine the people and organizations involved and the relevant locations:
  "Mrs. Green spoke today in New York. Green chairs the finance committee."
- Entities sometimes span multiple words. Entity of a word not obvious without considering its *context*
- CRF has one variable X_i for each word, which encodes the possible labels of that word
- The labels are, for example, "B-person, I-person, B-location, I-location, B-organization, I-organization"
  - Having beginning (B) and within (I) allows the model to segment adjacent entities

NLP example: named-entity recognition

The graphical model looks like (called a *skip-chain CRF*):



There are three types of potentials:

- $\phi^1(Y_t, Y_{t+1})$ represents dependencies between neighboring target variables [analogous to transition distribution in a HMM]
- $\phi^2(Y_t, Y_{t'})$ for all pairs t, t' such that $x_t = x_{t'}$, because if a word appears twice, it is likely to be the same entity
- $\phi^3(Y_t, X_1, \cdots, X_T)$ for dependencies between an entity and the word sequence [e.g., may have features taking into consideration capitalization]

Notice that the graph structure changes depending on the sentence!