Mayank Kejriwal

University of Southern California

Conditional Random Fields

Outline

- Modeling
- Inference
- Training
- Applications

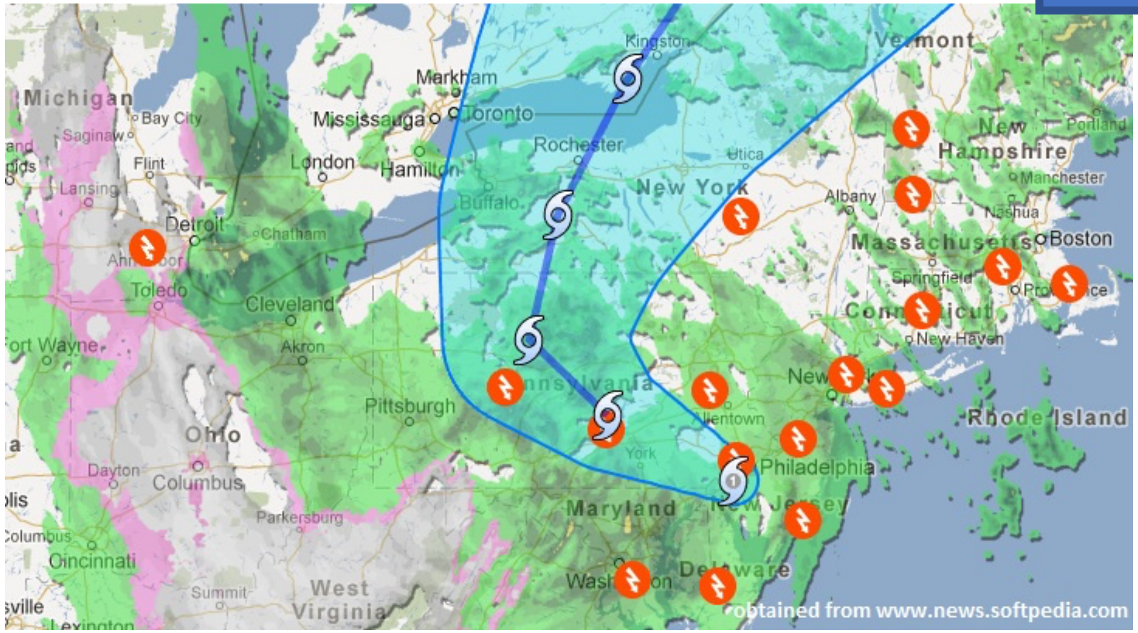
Problem Description

- Given X (observations), find Y (predictions)
- For example,

$$\begin{cases} X = \{temperature, moisture, pressure, \ldots\} \\ Y = \{Sunny, Rainy, Stormy, \ldots\} \end{cases}$$

Might depend on previous days and each other

Might depend on previous days and each other



Problem Description

- The relational connection occurs in many applications, NLP, Computer Vision, Signal Processing, ….
- Traditionally in graphical models, $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x})$
  - Modeling the joint distribution can lead to difficulties
  - rich local features occur in relational data, $p(\mathbf{x})$
  - features may have complex dependencies,
    - constructing probability distribution over them is difficult

Intractable!

- **Solution:** directly model the conditional, $p(\mathbf{y} \mid \mathbf{x})$
  - is sufficient for classification!
- **CRF** is simply a conditional distribution $p(\mathbf{y} \mid \mathbf{x})$ with an associated graphical structure

slide by Daniel Khashabi

Discriminative Vs. Generative

$p(\mathbf{y}, \mathbf{x})$

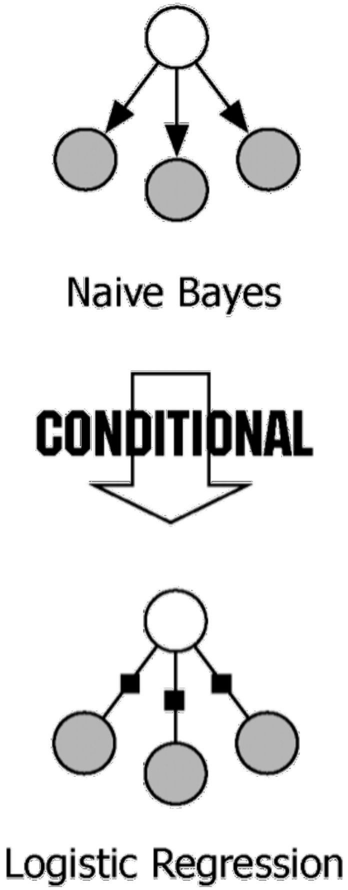
- **Generative Model:** A model that generate observed data randomly
- **Naïve Bayes:** once the class label is known, all the features are independent

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

$p(\mathbf{y} \mid \mathbf{x})$

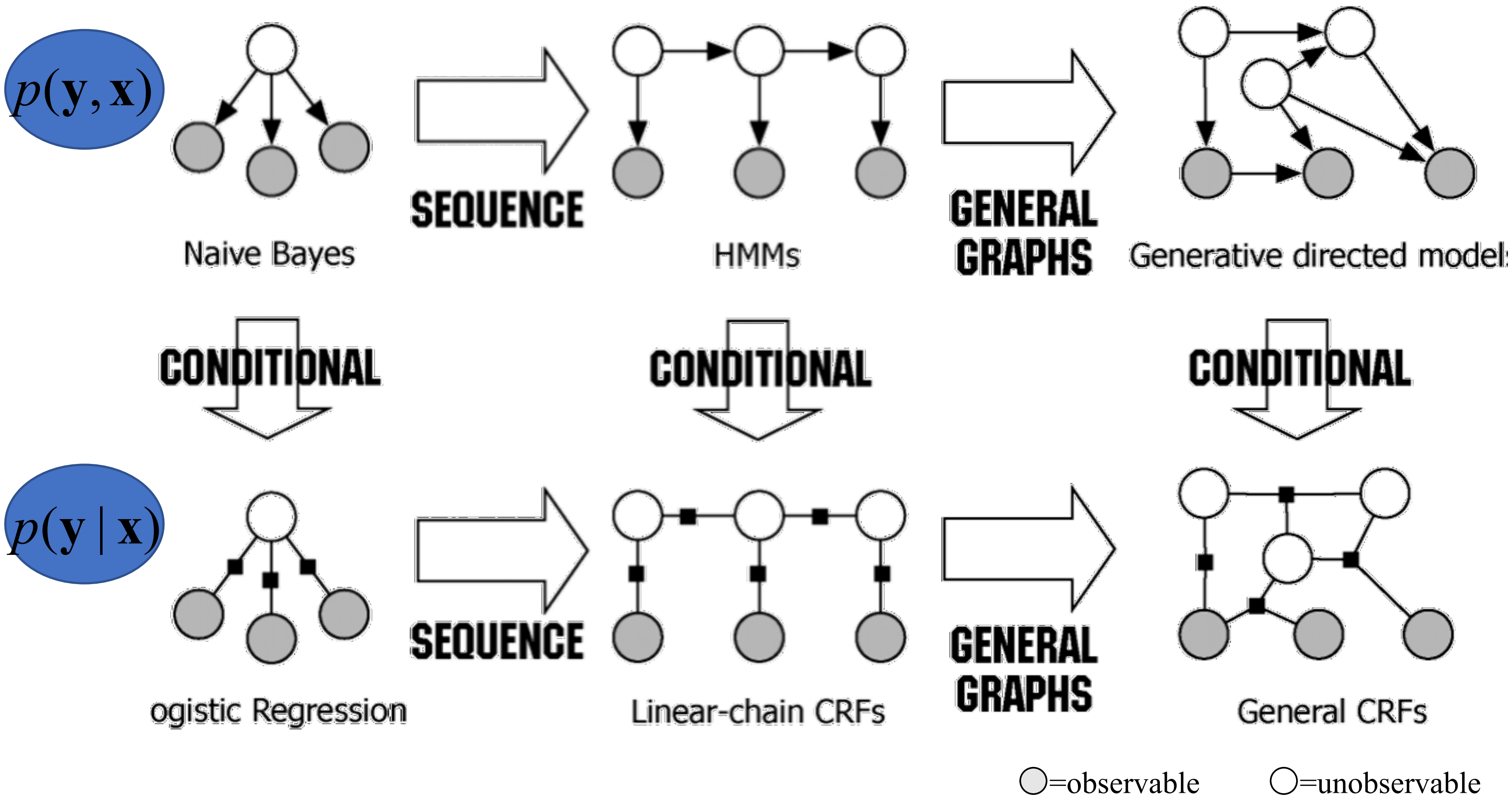
- **Discriminative:** Directly estimate the posterior probability; Aim at modeling the "discrimination" between different outputs
- **MaxEnt** classifier: linear combination of feature function in the exponent,

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y, \mathbf{x}) \right\}$$



Both generative models and discriminative models describe distributions over (y , x), but they work in different directions.

Discriminative Vs. Generative



slide by Daniel Khashabi

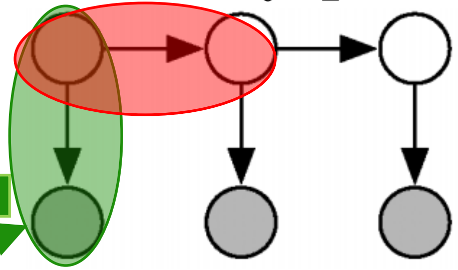
Sequence prediction

- Like NER: identifying and classifying proper names in text, e.g. China as location; George Bush as people; United Nations as organizations
  - Set of observation, $X = \{x_t\}_{t=1}^{T}$
  - Set of underlying sequence of states, $Y = \{y_t\}_{t=1}^{T}$
- HMM is generative:

Transition probability

$$p(\mathbf{y},\mathbf{x}) = \prod_{t=1} \boxed{p(y_t|y_{t-1})} \boxed{p(x_t|y_t)}$$

Observation probability



- Doesn’t model long-range dependencies
- Not practical to represent multiple interacting features (hard to model p(x))
- The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions
- And it can handle overlapping features

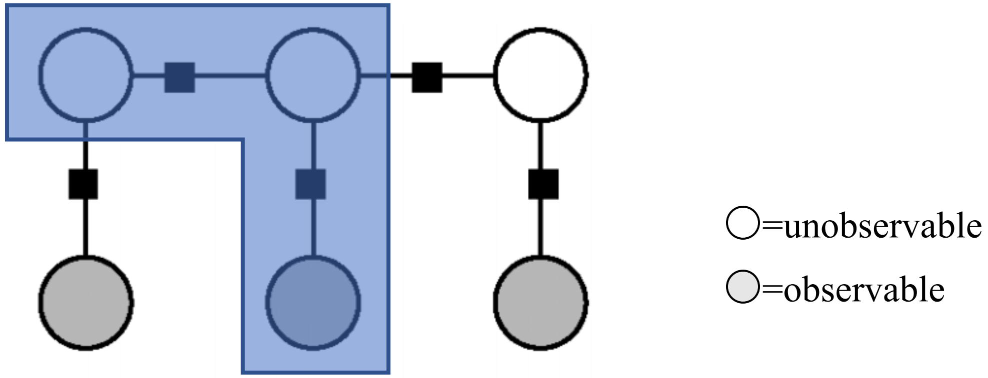
Chain CRFs

- Each potential function will operate on pairs of adjacent label variables

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_j \lambda_j F_j(\boldsymbol{y},\boldsymbol{x})\right)$$

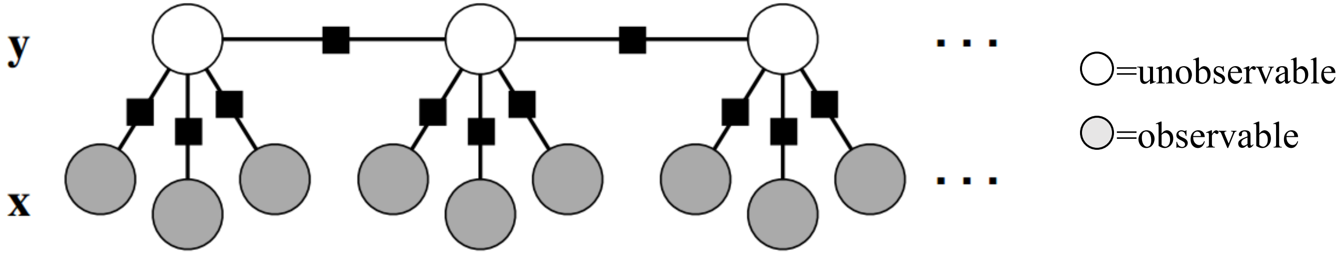
$$F_j(\boldsymbol{y},\boldsymbol{x}) = \sum_{i=1} \boxed{f_j(y_{i-1}, y_i, \boldsymbol{x}, i),} \Rightarrow \text{Feature functions}$$

- Parameters to be estimated, λ_j

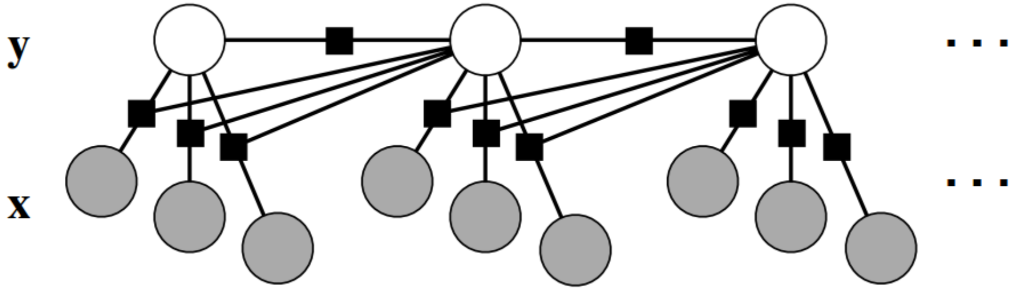


Chain CRF

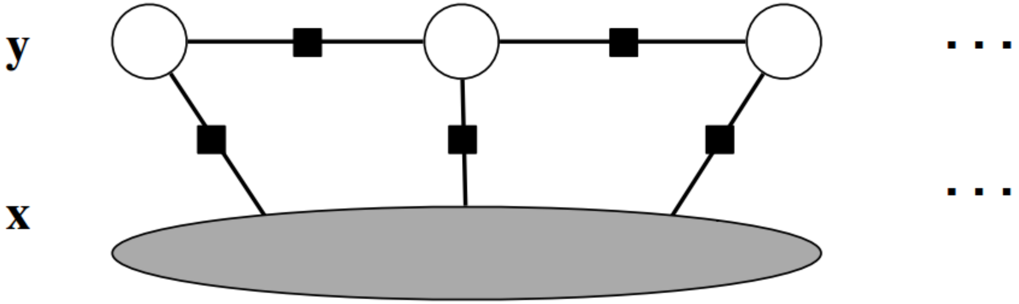
- We can change it so that each state depends on more observations



- Or inputs at previous steps



- Or all inputs

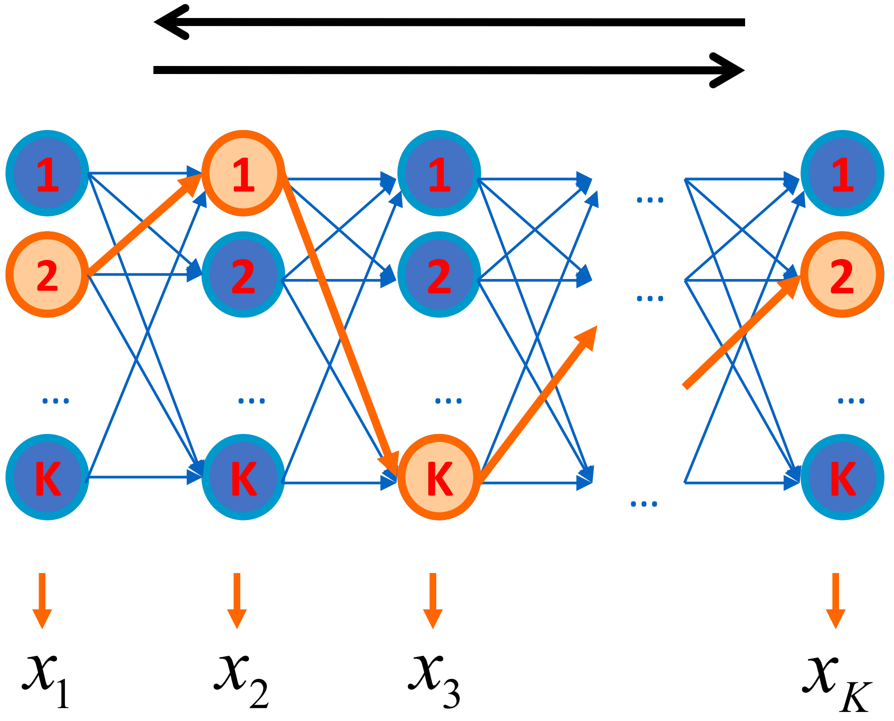


Outline

- Modeling
- Inference
- Training
- Applications

Inference in HMM

- Dynamic Programming:
  - Forward
  - Backward
  - Viterbi



Inference: Chain-CRF

- The inference of linear-chain CRF is very similar to that of HMM
- We can write the marginal distribution:

$$p(Y_{i-1} = y', Y_i = y | \boldsymbol{x}^{(k)}, \boldsymbol{\lambda}) = \frac{\alpha_{i-1}(y'|\boldsymbol{x}) M_i(y', y|\boldsymbol{x}) \beta_i(y|\boldsymbol{x})}{Z(\boldsymbol{x})}$$

- Solve Chain-CRF using Dynamic Programming (Similar to Viterbi)!
- 1. First computing *α* for all *t* (forward), then compute *β* for all *t* (backward).
- 2. Return the marginal distributions computed.
- 3. Run viterbi to find the optimal sequence

slide by Daniel Khashabi