Fall 2010 Graduate Course on
Dynamic Learning

Chapter 8: Conditional Random Fields

November 1, 2010

Byoung-Tak Zhang

School of Computer Science and Engineering &
Cognitive Science and Brain Science Programs
Seoul National University

http://bi.snu.ac.kr/~btzhang/

Overview

- Motivating Applications
  - Sequence Segmentation and Labeling
- Generative vs. Discriminative Models
  - HMM, MEMM
- CRF
  - From MRF to CRF
  - Learning Algorithms
- HMM vs. CRF

Motivating Application: Sequence Labeling

- Pos Tagging

  [He/PRP] [reckons/VBZ] [the/DT] [current/JJ] [account/NN] [deficit/NN] [will/MD] [narrow/VB] [to/TO] [only/RB] [#/#] [1.8/CD] [billion/CD] [in/IN] [September/NNP] [./.]

- Term Extraction

  Rockwell International Corp.'s Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to provide structural parts for Boeing's 747 jetliners.

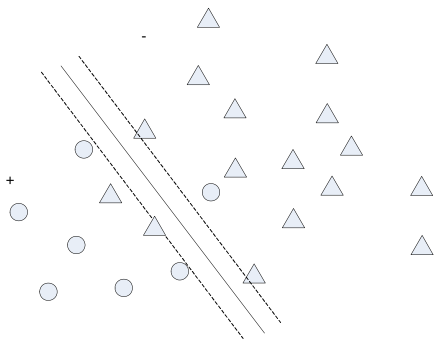
- Information Extraction from Company Annual Report

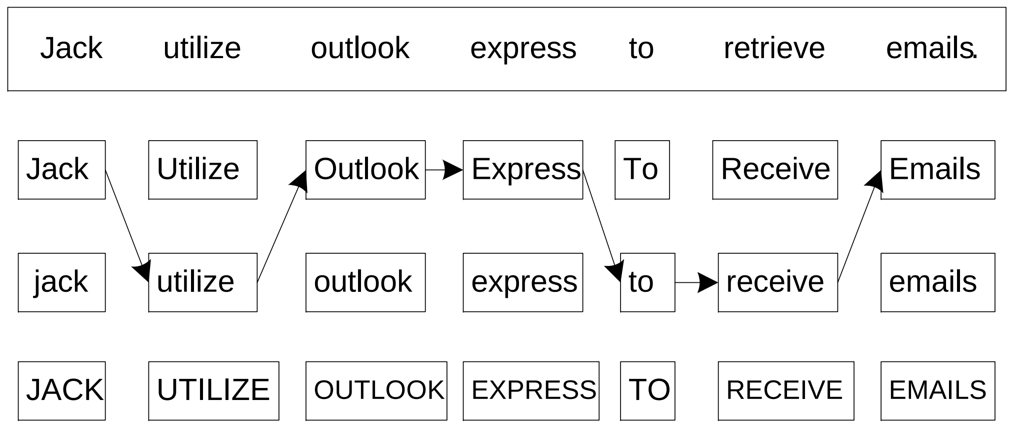
Sequence Segmenting and Labeling

- Goal: mark up sequences with content tags
- Computational linguistics
  - Text and speech processing
  - Topic segmentation
  - Part-of-speech (POS) tagging
  - Information extraction
  - Syntactic disambiguation
- Computational biology
  - DNA and protein sequence alignment
  - Sequence homolog searching in databases
  - Protein secondary structure prediction
  - RNA secondary structure analysis

Binary Classifier vs. Sequence Labeling

- Case restoration
  - jack utilize outlook express to retrieve emails
  - E.g. SVMs vs. CRFs





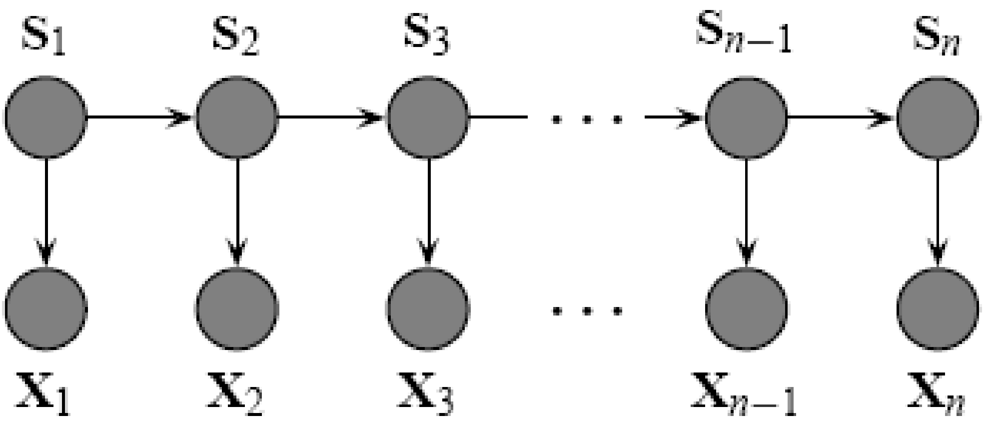
Sequence Labeling Models: Overview

- HMM
  - Generative model
  - E.g. Ghahramani (1997), Manning and Schutze (1999)
- MEMM
  - Conditional model
  - E.g. Berger and Pietra (1996), McCallum and Freitag (2000)
- CRFs
  - Conditional model without label bias problem
  - Linear-Chain CRFs
    - E.g. Lafferty and McCallum (2001), Wallach (2004)
  - Non-Linear Chain CRFs
    - Modeling more complex interaction between labels: DCRFs, 2D-CRFs
    - E.g. Sutton and McCallum (2004), Zhu and Nie (2005)

Generative Models: HMM

- Based on joint probability distribution $P(\mathbf{y},\mathbf{x})$
- Includes a model of $P(\mathbf{x})$ which is not needed for classification
- Interdependent features
  - either enhance model structure to represent them (→ complexity problems)
  - or make simplifying independence assumptions (e.g. naive Bayes)
- Hidden Markov models (HMMs) and stochastic grammars
  - Assign a joint probability to paired observation and label sequences
  - The parameters typically trained to maximize the joint likelihood of train examples

Hidden Markov Model



$$P(s,x) = P(s_1)P(x_1 \mid s_1)\prod_{i=2}^{n} P(s_i \mid s_{i-1})P(x_i \mid s_i)$$

Cannot represent multiple interacting (overlapping) features or long range dependences between observed elements.

Conditional Models

- Difficulties and disadvantages of generative models
  - Need to enumerate all possible observation sequences
  - Not practical to represent multiple interacting features or long-range dependencies of the observations
  - Very strict independence assumptions on the observations
- Conditional Models
  - Conditional probability $P(\mathbf{y}|\mathbf{x})$ rather than joint probability $P(\mathbf{y}, \mathbf{x})$ where y = label sequence and x = observation sequence.
  - Based directly on conditional probability $P(y|x)$
  - Need no model for $P(x)$
  - Specify the probability of possible label sequences given an observation sequence
  - Allow arbitrary, non-independent features on the observation sequence X
  - The probability of a transition between labels may depend on past and future observations
  - Relax strong independence assumptions in generative models

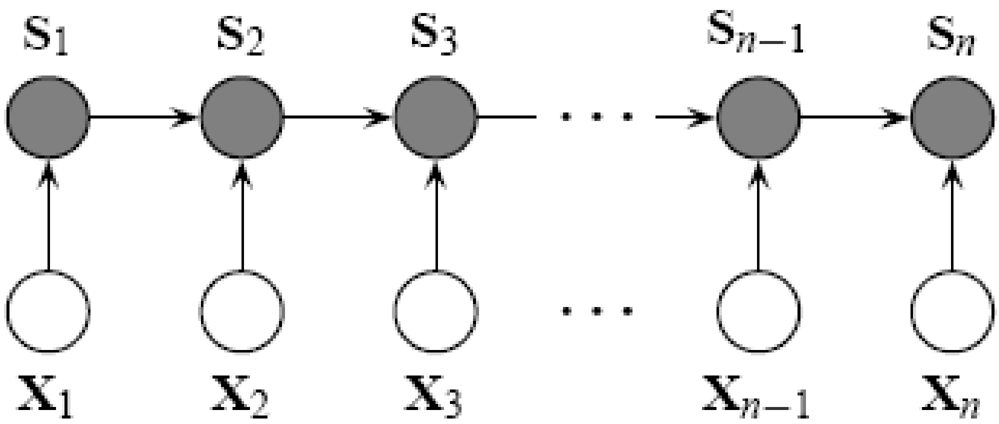
Discriminative Models: MEMM

- Maximum entropy Markov model (MEMM)
  - Exponential model
- Given a training set (X, Y) of observation sequences X and label sequences Y:
  - Train a model θ that maximizes $P(Y|X, \theta)$
  - For a new data sequence $\mathbf{x}$, the predicted label $\mathbf{y}$ maximizes $P(\mathbf{y}|\mathbf{x}, \theta)$
  - Notice the per-state normalization

$$P(y' \mid y, x) = \frac{1}{Z(y, x)} \exp\left(\sum_k \underbrace{\lambda_k}_{\text{weight}} \underbrace{f_k(x, y, y')}_{\text{feature}}\right)$$

- MEMMs have all the advantages of conditional models
- Per-state normalization: all the mass that arrives at a state must be distributed among the possible successor states ("conservation of score mass")
- Subject to Label Bias Problem
  - Bias toward states with fewer outgoing transitions

Maximum Entropy Markov Model

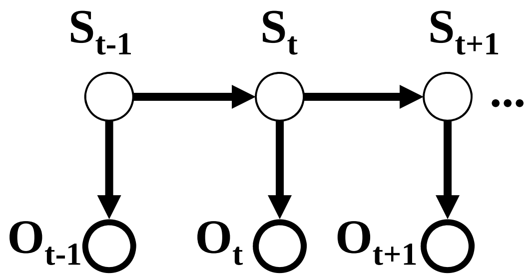


$$P(s \mid x) = P(s_1 \mid x_1) \prod_{i=2}^{n} P(s_i \mid s_{i-1}, x_i)$$

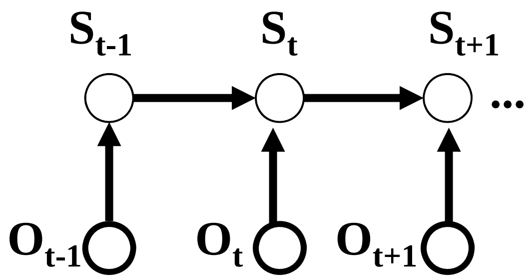
Label bias problem: the probability transitions leaving any given state must sum to one

Conditional Markov Models (CMMs) aka MEMMs aka Maxent Taggers *vs.* HMMs

$$P(s,o)=\prod_i P(s_i \mid s_{i-1})P(o_i \mid s_{i-1})$$



$$P(s \mid o)=\prod_i P(s_i \mid s_{i-1}, o_{i-1})$$



MEMM to CRFs

$$P(y_1 \ldots y_n \mid x_1 \ldots x_n) = \prod_j P(y_j \mid y_{j-1,} x_j) = \prod_j \frac{\exp(\sum_i \lambda_i f_i(x_j, y_j, y_{j-1}))}{Z_\lambda(x_j)}$$

$$= \frac{\exp(\sum_i \lambda_i F_i(\mathbf{x}, \mathbf{y}))}{\prod_j Z_\lambda(x_j)}, \text{ where } F_i(\mathbf{x}, \mathbf{y}) = \sum_j f_i(x_j, y_j, y_{j-1})$$

New model

$$\frac{\exp(\sum_i \lambda_i F_i(\mathbf{x}, \mathbf{y}))}{\boxed{Z_\lambda(\mathbf{x})}}$$

HMM, MEMM, and CRF in Comparison

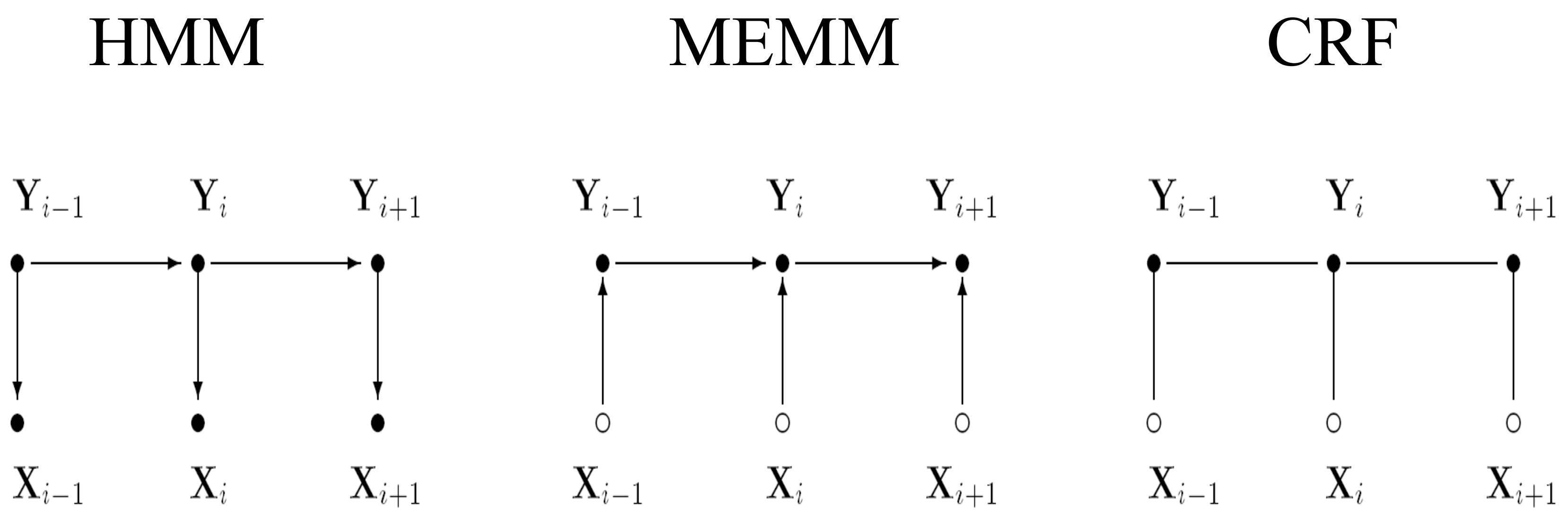


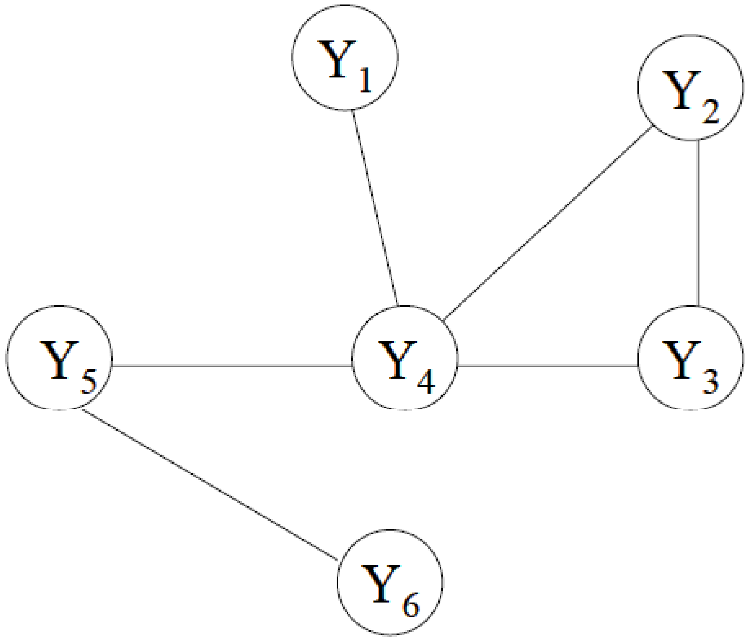
Figure 2. Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model.

Conditional Random Field (CRF)

Random Field

Let $G = (Y, E)$ be a graph where each vertex Y_v is a random variable
Suppose $P(Y_v \mid \text{all other } Y) = P(Y_v \mid \text{neighbors}(Y_v))$ then Y is a random field

Example:



- $P(Y_5 \mid \text{all other } Y) = P(Y_5 \mid Y_4, Y_6)$

Markov Random Field

- **Random Field**: Let $F = \{F_1, F_2, \ldots, F_M\}$ 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 it satisfies the Markov property.
  - undirected graph for joint probability p(x)
  - allows no direct probabilistic interpretation
  - define potential functions Ψ on maximal cliques A
    - map joint assignment to non-negative real number
    - requires normalisation

$$p(x) = \frac{1}{Z} \prod_{A} \Psi_A(x_A) \qquad Z = \sum_{x} \prod_{A} \Psi_A(x_A)$$

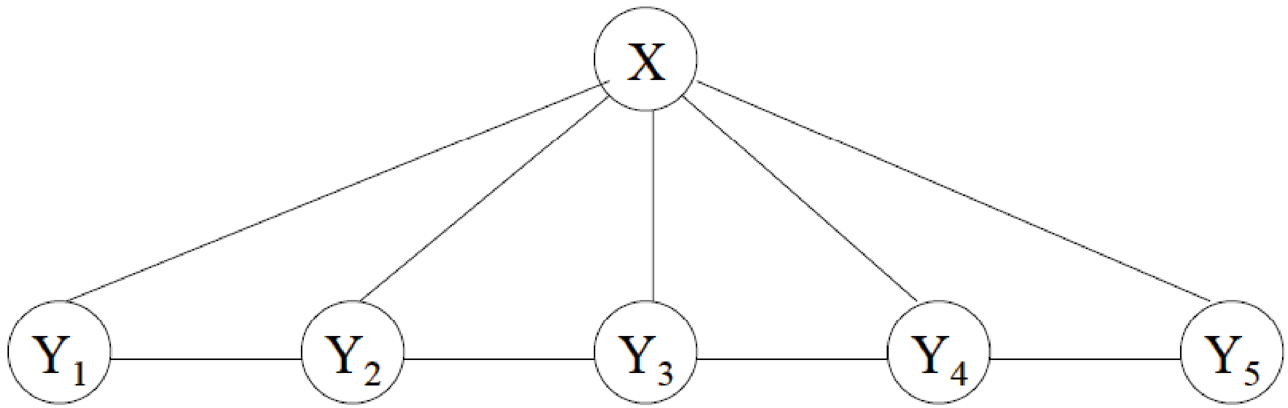
Conditional Random Field: CRF

- Conditional probabilistic sequential models $p(\mathrm{y}|\mathrm{x})$
- Undirected graphical models
- Joint probability of an entire label sequence given a particular observation sequence
- Weights of different features at different states can be traded off against each other

$$p(y \mid x) = \frac{1}{Z} \prod_{A} \Psi_A(x_A, y_A) \qquad Z(x) = \sum_{y} \prod_{A} \Psi_A(x_A, y_A)$$

Example of CRFs

Suppose $P(Y_v \mid X, \text{all other } Y) = P(Y_v \mid X, \text{neighbors}(Y_v))$
then X with Y is a **conditional** random field



- $P(Y_3 \mid X, \text{all other } Y) = P(Y_3 \mid X, Y_2, Y_4)$
- Think of X as observations and Y as labels

Definition of CRFs

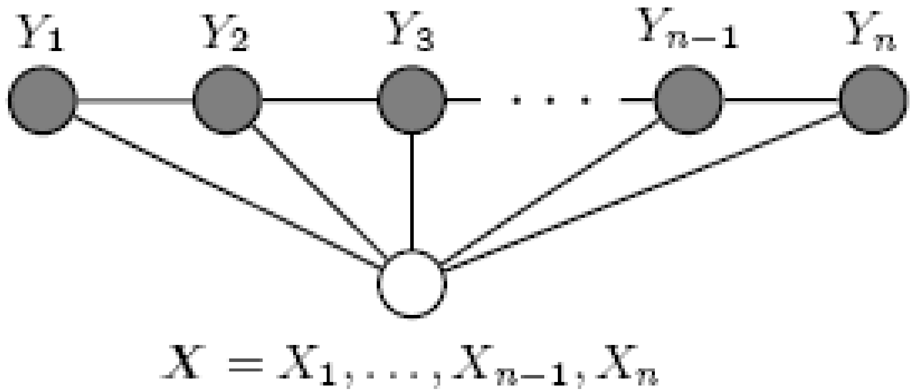
X is a random variable over data sequences to be labeled
Y is a random variable over corresponding label sequences

Definition. *Let* $G = (V, E)$ *be a graph such that* $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, *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 Markov property with respect to the graph:* $p(\mathbf{Y}_v \mid \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v \mid \mathbf{X}, \mathbf{Y}_w, w \sim v)$, *where* $w \sim v$ *means that* w *and* v *are neighbors in* G.

Conditional Random Fields (CRFs)

- 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
- Undirected acyclic graph
- Allow some transitions “vote” more strongly than others depending on the corresponding observations

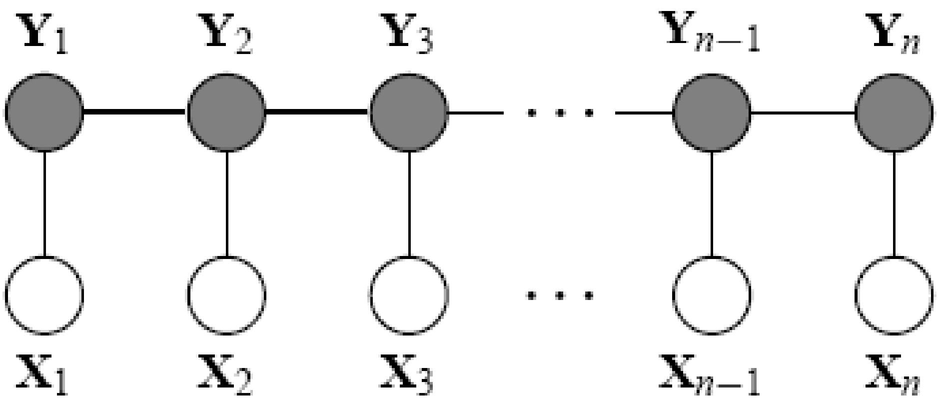
Conditional Random Field



Graphical structure of a chain-structured CRFs for sequences. The variables corresponding to unshaded nodes are not generated by the model.

Conditional Random Field: a Markov random field (*Y*) globally conditioned on another random field (*X*).

Conditional Random Field



undirected graphical model globally conditioned on X

Given an undirected graph $G = (V, E)$ such that $Y = \{Y_v \mid v \in V\}$, if

$$p(Y_v \mid X, Y_u, u \neq v, \{u, v\} \in V) \Leftrightarrow p(Y_v \mid X, Y_u, (u, v) \in E)$$

The probability of Y_v given X and those random variables corresponding to nodes neighboring v in G. Then (X, Y) is a conditional random field.

Conditional Distribution

If the graph $G = (V, E)$ of Y is a tree, the conditional distribution over the label sequence Y = y, given X = x, by fundamental theorem of random fields is:

$$p_\theta(\mathrm{y} \mid \mathrm{x}) \propto \exp\left(\sum_{e \in E, k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V, k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

x is a data sequence

y is a label sequence

v is a vertex from vertex set V = set of label random variables

e is an edge from edge set E over V

f_k and g_k are given and fixed. g_k is a Boolean vertex feature; f_k is a Boolean edge feature

k is the number of features

$\theta = (\lambda_1, \lambda_2, \cdots, \lambda_n; \mu_1, \mu_2, \cdots, \mu_n)$; λ_k and μ_k are parameters to be estimated

$\mathrm{y}|_e$ is the set of components of y defined by edge e

$\mathrm{y}|_v$ is the set of components of y defined by vertex v

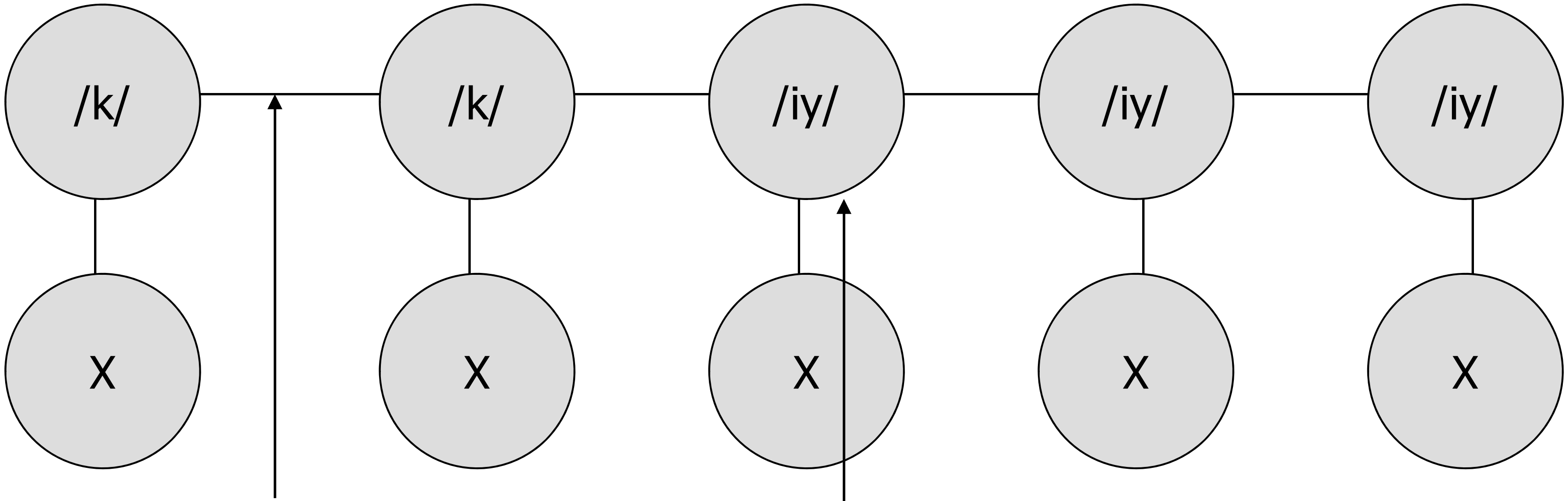
Conditional Distribution (cont'd)

- CRFs use the observation-dependent normalization $Z(\mathrm{x})$ for the conditional distributions:

$$p_\theta(\mathrm{y} \mid \mathrm{x}) = \frac{1}{Z(\mathrm{x})} \exp\left(\sum_{e \in E, k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V, k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

$Z(\mathrm{x})$ is a normalization over the data sequence x

Conditional Random Fields



Transition functions add associations between transitions f one label to anothe

State functions help determine the identity of the state

- CRFs larkov Random Fields
  - Mo observations
  - Observations in a CRF ar variables

Conditional Random Fields

$$P(y|x) = \frac{\exp \sum_t \left(\sum_i \lambda_i f_i(x, y_t) + \sum_j \mu_j g_j(x, y_t, y_{t-1}) \right)}{Z(x)}$$

- Hammersley-Clifford Theorem states that a random field is an MRF iff it can be described in the above form
- The exponential is the sum of the *clique potentials* of the undirected graph

State Feature Weig

λ=10

One possible weight va

for this state feature

(Strong)

Transition Feature Weight

μ=4

One possible weight value

for this transition feature

Transition Feature Function

g(x, /iy/,/k/)

One possible transition feature function

Indicates /k/ followed by /iy/

Conditional Random Fields

- Each attribute of the data we are trying to model fits into a *feature function* that associates the attribute and a possible label
  - A positive value if the attribute appears in the data
  - A zero value if the attribute is not in the data
- Each feature function carries a *weight* that gives the strength of that feature function for the proposed label
  - High positive weights indicate a good association between the feature and the proposed label
  - High negative weights indicate a negative association between the feature and the proposed label
  - Weights close to zero indicate the feature has little or no impact on the identity of the label

Formally …. Definition

- CRF is a Markov random field.
- By the Hammersley-Clifford theorem, the probability of a label can be expressed as a Gibbs distribution, so that

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j F_j(y, x))$$

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{|c}, x, i)$$

clique

- What is clique?
- By only taking consideration of the one-node and two-node cliques, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j t_j(y_{|e}, x, i) + \sum_k \mu_k s_k(y_{|s}, x, i))$$

Definition (cont.)

Moreover, let us consider the problem in a first-order chain model, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\Big(\sum_j \lambda_j t_j (y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k (y_i, x, i)\Big)$$

For simplifying description, let $f_j(y, x)$ denote $t_j(y_{i-1}, y_i, x, i)$ and $s_k(y_i, x, i)$

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\Big(\sum_j \lambda_j F_j (y, x)\Big)$$

$$F_j (y, x) = \sum_{i=1}^{n} f_j (y_{|c}, x, i)$$

Labeling

- In labeling, the task is to find the label sequence that has the largest probability
- Then the key is to estimate the parameter lambda

$$\hat{y} = \arg\max_{y} p_{\lambda}(y \mid x) = \arg\max_{y} (\lambda \cdot F(y, x))$$

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_{j} \lambda_j F_j(y, x))$$

Optimization

- Defining a loss function that should be convex for avoiding local optimization
- Defining constraints
- Finding a optimization method to solve the loss function
- A formal expression for optimization problem

$$
\begin{aligned}
&\min_{\theta} \ f(x) \\
&s.t. \qquad g_i(x) \geq 0, 0 \leq i \leq k \\
&\qquad\qquad h_j(x) = 0, 0 \leq j \leq l
\end{aligned}
$$

Loss Function

Empirical loss vs. structural loss

$$\text{minimize} \quad L = \sum_k \left| y - f(x, \lambda) \right|$$

$$\text{minimize} \quad L = \|\lambda\| + \sum_k \left| y - f(x, \lambda) \right|$$

Loss function: Log-likelihood

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

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

$$L_\lambda = \sum_k \left[\lambda \cdot F(y^{(k)}, x^{(k)}) - \log Z_\lambda(x^{(k)}) \right] - \frac{\|\lambda\|^2}{2\sigma^2} + const$$

Parameter Estimation

Log-likelihood

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

Differentiating the log-likelihood with respect to parameter λ_j

$$\frac{\delta L}{\delta \lambda_j} = E_{p(Y,X)}[F_j(Y,X)] - \sum_k E_{p(Y|x^{(k)},\lambda)}[F_j(Y,x^{(k)})]$$

By adding the model penalty, it can be rewritten as

$$\frac{\delta L}{\delta \lambda_j} = E_{p(Y,X)}[F_j(Y,X)] - \sum_k E_{p(Y|x^{(k)},\lambda)}[F_j(Y,x^{(k)})] - \frac{\lambda}{\sigma^2}$$

$$\frac{\delta L_\lambda}{\delta \lambda_j} = \sum_k \left[F_j(y^{(k)}, x^{(k)}) - \frac{(Z_\lambda(x^{(k)}))'}{Z_\lambda(x^{(k)})} \right]$$

$$Z_\lambda(x^{(k)}) = \sum_y \exp \lambda \cdot F(y, x^{(k)})$$

$$\frac{(Z_\lambda(x^{(k)}))'}{Z_\lambda(x^{(k)})} = \frac{\sum_y \left(\exp(\lambda \cdot F(y,x^{(k)})) * F_j(y,x^{(k)}) \right)}{\sum_y \exp \lambda \cdot F(y,x^{(k)})}$$

$$= \sum_y \left(\frac{\exp(\lambda \cdot F(y,x^{(k)}))}{\sum_y \exp \lambda \cdot F(y,x^{(k)})}) * F_j(y,x^{(k)}) \right)$$

$$= \sum_y \left(p(y \mid x^{(k)}) * F_j(y,x^{(k)}) \right)$$

$$= E_{p(Y|x^{(k)})} F_j(Y, x^{(k)})$$

Optimization

$$L(\lambda) = \sum_k \left[-\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

$$\frac{\delta L}{\delta \lambda_j} = \mathbf{E}_{p(Y,X)}[F_j(Y,X)] - \sum_k \mathbf{E}_{p(Y|x^{(k)},\lambda)}[F_j(Y, x^{(k)})]$$

- $\mathbf{E}_{p(y,x)} F_j(y,x)$ can be calculated easily
- $\mathbf{E}_{p(y|x)} F_j(y,x)$ can be calculated by making use of a forward-backward algorithm
- Z can be estimated in the forward-backward algorithm

Calculating the Expectation

- First we define the transition matrix of y for position x as

$$M_i[y_{i-1}, y_i] = \exp \lambda \cdot f(y_{i-1}, y_i, x, i)$$

$$E_{p_\lambda(Y|x^{(k)})}\left[F_j(Y, x^{(k)})\right] = \sum_y p_\lambda(y \mid x^{(k)}) F_j(y, x)$$

$$= \sum_{i=1}^{n} \sum_{y_{i-1}, y_i} p(y_{i-1}, y_i \mid x^{(k)}) f_j(y_{i-1}, y_i, x^{(k)})$$

$$= \sum_i \frac{\alpha_{i-1}(f_i * M_i * V_i)\beta_i^T}{Z_\lambda(x)}$$

$$Z_\lambda(x) = \left[\prod_{i=1}^{n+1} M_i(x)\right] = \alpha_n \cdot 1^T$$

$$\alpha_i = \begin{cases} \alpha_i M_i & 0 < i \le n \\ 1 & i = 0 \end{cases}$$

$$\beta_i^T = \begin{cases} M_{i+1}\beta_{i+1}^T & 1 \le i < n \\ 1 & i = n \end{cases}$$

$$p(y_i \mid x^{(k)}) = \frac{\alpha_{i-1}\beta_i^T}{Z_\lambda(x)}$$

All state features at position *i*

First-order Numerical Optimization

Using Iterative Scaling (GIS, IIS)

- Initialize each λ_j (= 0 for example)
- Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_j} = 0$ for each parameter λ_j
  - Update each parameter using $\lambda_j \leftarrow \lambda_j + \Delta\lambda_j$

Second-order Numerical Optimization

Using newton optimization technique for the parameter estimation

$$\lambda^{(k+1)} = \lambda^{(k)} + \left(\frac{\partial^2 L}{\partial \lambda^2}\right)^{-1} \frac{\partial L}{\partial \lambda}$$

Drawbacks: parameter value initialization
And compute the second order (i.e. Hesse matrix), that is difficult

Solutions:

- Conjugate-gradient (CG) (Shewchuk, 1994)
- Limited-memory quasi-Newton (L-BFGS) (Nocedal and Wright, 1999)
- Voted Perceptron (Colloins 2002)

Summary of CRFs

Model
- Lafferty, 2001

Applications
- Efficient training (Wallach, 2003)
- Training via. Gradient Tree Boosting (Dietterich, 2004)
- Bayesian Conditional Random Fields (Qi, 2005)
- Name entity (McCallum, 2003)
- Shallow parsing (Sha, 2003)
- Table extraction (Pinto, 2003)
- Signature extraction (Kristjansson, 2004)
- Accurate Information Extraction from Research Papers (Peng, 2004)
- Object Recognition (Quattoni, 2004)
- Identify Biomedical Named Entities (Tsai, 2005)
- …

Limitation
- Huge computational cost in parameter estimation

HMM vs. CRF

HMM

$$\arg\max_{\phi} P(\phi \mid S)$$

$$= \arg\max_{\phi} P(\phi)P(S \mid \phi)$$

$$= \arg\max_{\phi} \sum_{y_i \in \phi} \log\left(P_{trans}(y_i \mid y_{i-1})P_{emit}(s_i \mid y_i)\right)$$

CRF

$$\arg\max_{\phi} P(\phi \mid S)$$

$$= \arg\max_{\phi} \frac{1}{Z} e^{\sum \lambda f(c,S)}$$

$$= \arg\max_{\phi} \sum_{c,i} \lambda_i f_i(c,S)$$

1. Both optimizations are over *sums*—this allows us to use any of the dynamic programming HMM/GHMM decoding algorithms for fast, memory-efficient parsing, with the CRF scoring scheme used in place of the HMM/GHMM scoring scheme.

2. The CRF functions $f_i(c,S)$ may in fact be implemented using any type of sensor, including such *probabilistic sensors* as Markov chains, interpolated Markov models (IMM's), decision trees, phylogenetic models, etc..., as well as any *non-probabilistic* sensor, such as n-mer counts or binary indicators.

Appendix

Markov Random Fields

A (discrete-valued) *Markov random field* (*MRF*) is a 4-tuple $M=(\alpha, X, P_M, G)$ where:

- α is a finite *alphabet*,
- X is a set of (observable or unobservable) *variables* taking values from α,
- P_M is a *probability distribution* on variables in X,
- $G=(X, E)$ is an undirected graph on X describing a set of *dependence relations* among variables,

such that $P_M(X_i|\{X_{k\neq i}\}) = P_M(X_i|\mathcal{N}_G(X_i))$, for $\mathcal{N}_G(X_i)$ the neighbors of X_i under G.

That is, the conditional probabilities as given by P_M must obey the dependence relations (a generalized "Markov assumption") given by the undirected graph G.

A problem arises when actually inducing such a model in practice—namely, that we can't just set the conditional probabilities $P_M(X_i \mid \mathcal{N}_G(X_i))$ arbitrarily and expect the joint probability $P_M(X)$ to be well-defined (Besag, 1974).

Thus, the problem of estimating parameters locally for each neighborhood is confounded by constraints at the global level...

Duke
UNIVERSITY

The Hammersley-Clifford Theorem

Suppose $P(\mathbf{x})>0$ for all (joint) value assignments $\mathbf{x}$ to the variables in X. Then by the Hammersley-Clifford theorem, the likelihood of $\mathbf{x}$ under model M is given by:

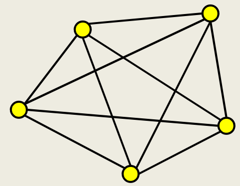
$$P_M(\mathbf{x}) = \frac{1}{Z} e^{Q(\mathbf{x})}$$

for normalization term Z:

$$Z = \sum_{\mathbf{x}'} e^{Q(\mathbf{x}')}$$

where $Q(\mathbf{x})$ has a unique expansion given by:

What is a clique?



A clique is any subgraph in which all vertices are neighbors.

$$Q(x_0, x_1, \ldots, x_{n-1}) = \sum_{0 \le i < n} x_i \Phi_i(x_i) + \sum_{0 \le i < j < n} x_i x_j \Phi_{i,j}(x_i, x_j) + \ldots$$
$$\ldots + x_0 x_1 \ldots x_{n-1} \Phi_{0,1,\ldots,n-1}(x_0, x_1, \ldots, x_{n-1})$$

and where any Φ_i term not corresponding to a *clique* must be zero. (Besag, 1974)

The reason this is useful is that it provides a way to evaluate probabilities (whether joint or conditional) based on the "local" functions Φ.

Thus, we can train an MRF by learning individual Φ functions—one for each clique.

Conditional Random Fields

A *Conditional random field* (*CRF*) is a *Markov random field* of *unobservables* which are globally conditioned on a set of *observables* (Lafferty *et al.*, 2001):

Formally, a CRF is a 6-tuple $M=(L,\alpha,Y,X,\Omega,G)$ where:

- L is a finite *output alphabet* of *labels*; e.g., {`exon`, `intron`},
- α is a finite *input alphabet* e.g., {A, C, G, T},
- Y is a set of *unobserved variables* taking values from L,
- X is a set of (fixed) *observed variables* taking values from α,
- $\Omega = \{\Phi_c : L^{|Y|} \times \alpha^{|X|} \rightarrow \mathbb{R}\}$ is a set of *potential functions*, $\Phi_c(\mathbf{y},\mathbf{x})$,
- $G=(V, E)$ is an undirected graph describing a set of *dependence relations* E among variables $V = X \cup Y$, where $E \cap (X \times X)=\varnothing$,

such that $(\alpha, Y, e^{\Sigma\Phi(c,\mathbf{x})}/Z, G-X)$ is a Markov random field.

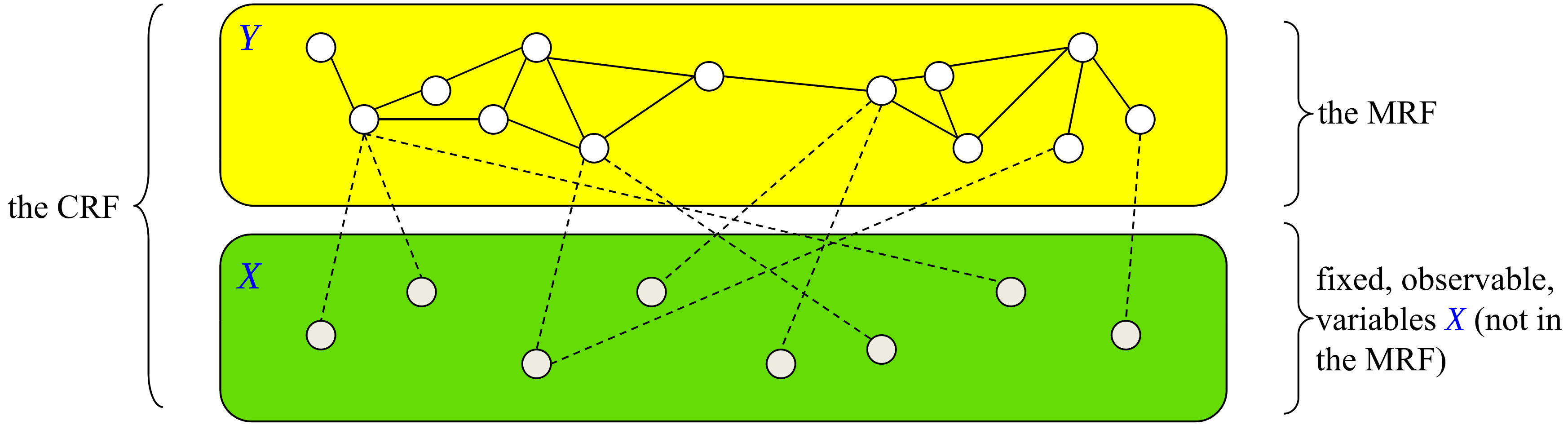
Note that:

1. The observables X are not included in the MRF part of the CRF, which is only over the subgraph $G-X$. However, the X are deemed *constants*, and are *globally visible* to the Φ functions.

2. We have not specified a probability function P_M, but have instead given "local" *clique-specific* functions Φ_c which together define a coherent probability distribution via Hammersley-Clifford.

CRF's versus MRF's

A conditional random field is effectively an MRF plus a set of "external" variables X, where the "internal" variables Y of the MRF are the unobservables (○) and the "external" variables X are the observables (●):



Thus, we could denote a CRF informally as:

$$C=(M, X)$$

for MRF M and external variables X, with the understanding that the graph $G_{X\cup Y}$ of the CRF is simply the graph G_Y of the underlying MRF M plus the vertices X and any edges connecting these to the elements of G_Y.

Note that in a CRF *we do not explicitly model any direct relationships between the observables (i.e., among the X)* (Lafferty *et al*., 2001).

Duke UNIVERSITY