CSE574 - Administriva

- No class on Fri 01/25 (Ski Day)

Last Wednesday

- **HMMs**
  - Most likely individual state at time t: (forward)
  - Most likely sequence of states (Viterbi)
  - Learning using EM
- **Generative vs. Discriminative Learning**
  - Model p(y,x) vs. p(y|x)
  - p(y|x) : don't bother about p(x) if we only want to do classification

Today

- **Markov Networks**
  - Most likely individual state at time t: (forward)
  - Most likely sequence of states (Viterbi)
  - Learning using EM
- **CRFs**
  - Model p(y,x) vs. p(y|x)
  - p(y|x) : don't bother about p(x) if we only want to do classification

Finite State Models

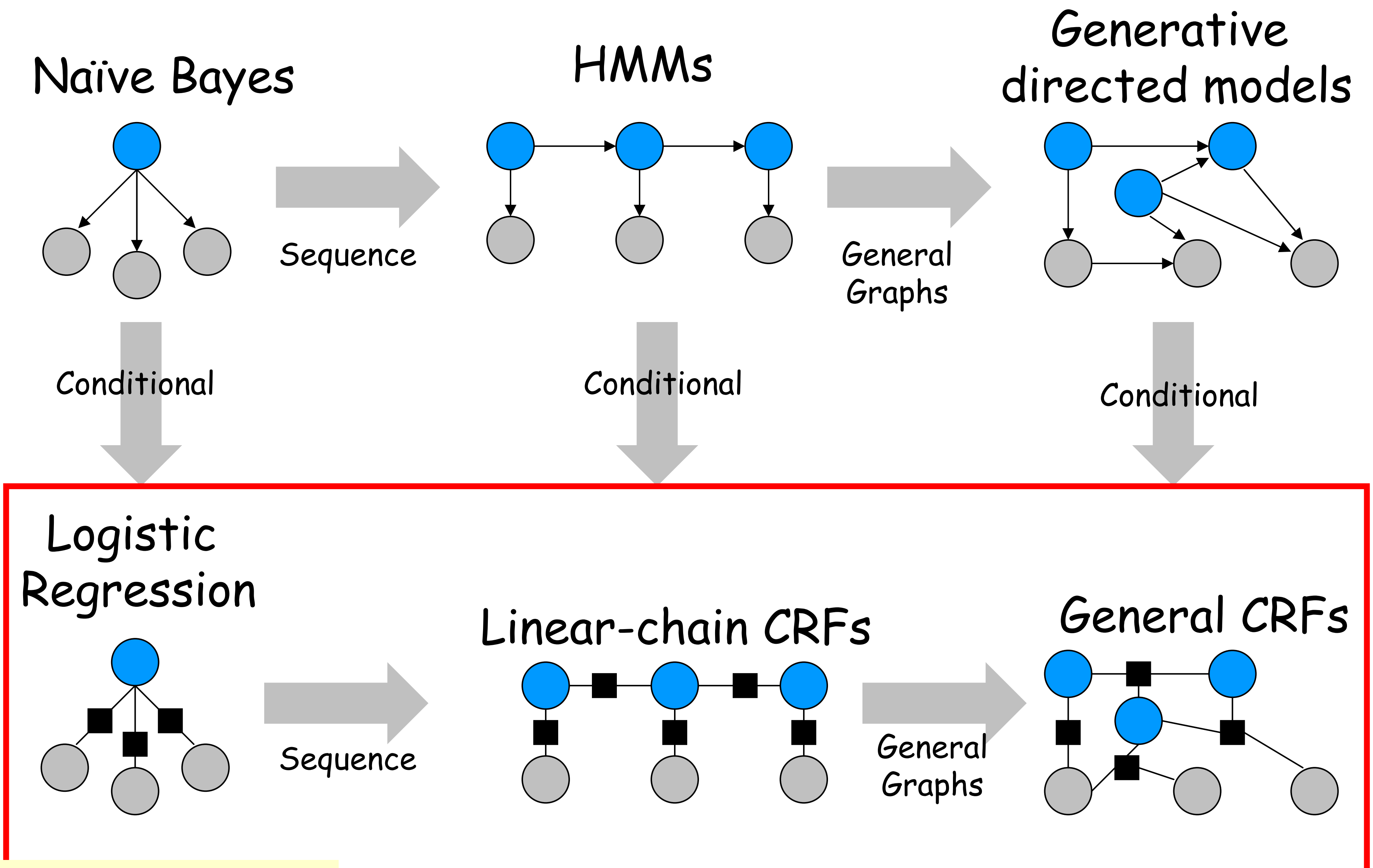


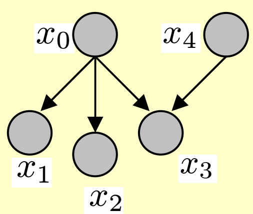
Figure by Sutton & McCallum

Graphical Models

- Family of probability ... certain way

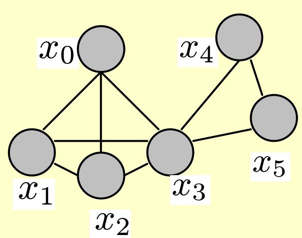
> Node is independent of its non-descendants given its parents

- Directed (Bayes Nets)



> Node is independent all other nodes given its neighbors

- Undirected (Markov Random Field)

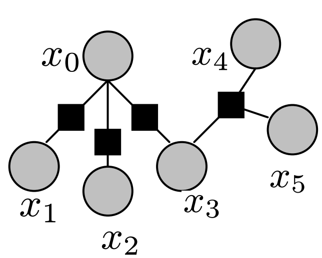


$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \Psi_C(\mathbf{x}_C)$$

$C \subset \{x_1, \ldots, x_K\}$ clique

Ψ_C potential function

- Factor Graphs



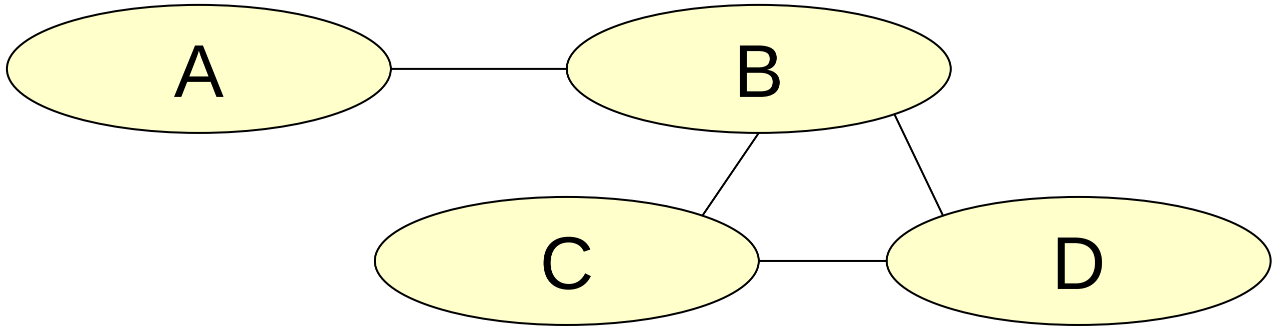
$$p(\mathbf{x}) = \frac{1}{Z} \prod_A \Psi_A(\mathbf{x}_A)$$

$A \subset \{x_1, \ldots, x_K\}$

Ψ_A factor function

Markov Networks

- Undirected **graphical models**



- **Potential functions defined over cliques**

$$P(X) = \frac{1}{Z} \prod_{c} \Phi_{c}(X) \qquad Z = \sum_{X} \prod_{c} \Phi_{c}(X)$$

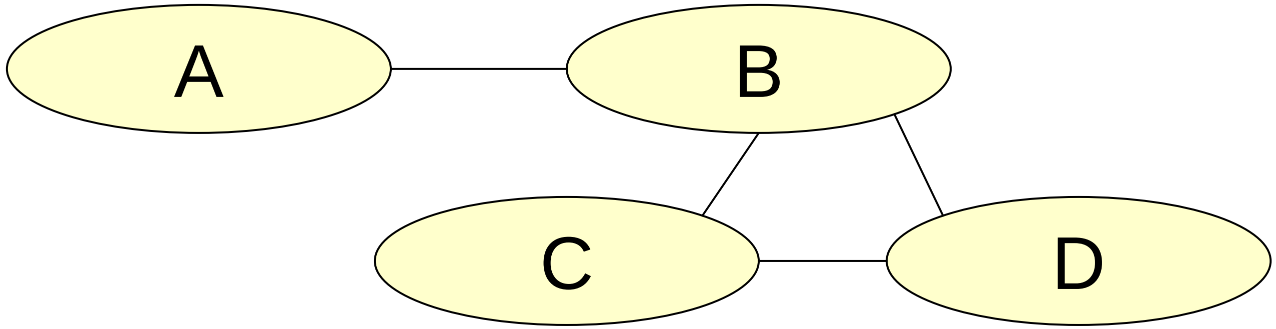
$$\Phi(A,B) = \begin{cases} 3.7 & \text{if A and B} \\ 2.1 & \text{if A and } \overline{\text{B}} \\ 0.7 & \text{otherwise} \end{cases}$$

$$\Phi(B,C,D) = \begin{cases} 2.3 & \text{if B and } \overline{\text{C}} \text{ and D} \\ 5.1 & \text{otherwise} \end{cases}$$

Slide by Domingos

Markov Networks

- Undirected **graphical models**



- **Potential functions defined over cliques**

$$P(X) = \frac{1}{Z}\exp\left(\sum_i \boxed{w_i}\,\boxed{f_i(X)}\right) \qquad Z = \sum_X \exp\left(\sum_i w_i f_i(X)\right)$$

Weight of Feature *i*

Feature *i*

$$f(A,B) = \begin{cases} 1 & \text{if A and B} \\ 0 & \text{otherwise} \end{cases}$$

$$f(B,C,D) = \begin{cases} 1 & \text{if B and } \overline{\text{C}} \text{ and D} \\ 0 & \end{cases}$$

Hammersley-Clifford Theorem

If Distribution is strictly positive ($P(x) > 0$)

And Graph encodes conditional independences

Then Distribution is product of potentials over cliques of graph

Inverse is also true.

Markov Nets vs. Bayes Nets

Property	Markov Nets	Bayes Nets
Form	Prod. potentials	Prod. potentials
Potentials	Arbitrary	Cond. probabilities
Cycles	Allowed	Forbidden
Partition func.	Z = ?	Z = 1
Indep. check	Graph separation	D-separation
Indep. props.	Some	Some
Inference	MCMC, BP, etc.	Convert to Markov

Slide by Domingos

Inference in Markov Networks

- **Goal: compute marginals & conditionals of**

$$P(X) = \frac{1}{Z}\exp\left(\sum_i w_i f_i(X)\right) \qquad = \sum \exp\left(\sum_i w_i f_i(X)\right)$$

- **Exa**
- **Con** **:**

E.g.: What is $P(x_i)$?

What is $P(x_i|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_N)$?

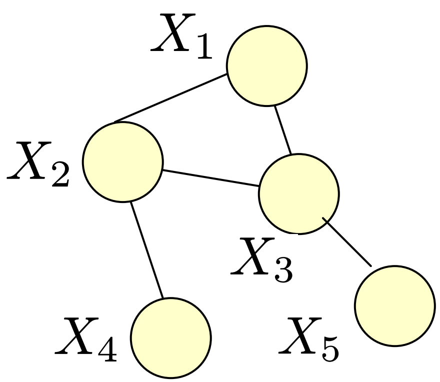
$$P(x \mid MB(x)) = \frac{\exp\left(\sum_i w_i f_i(x)\right)}{\exp\left(\sum_i w_i f_i(x=0)\right) + \exp\left(\sum_i w_i f_i(x=1)\right)}$$

- **Gibbs sampling exploits this**

Slide by Domingos

Markov Chain Monte Carlo

- **Idea:**
  - create chain of samples $x^{(1)}, x^{(2)}, \ldots$ where $x(i+1)$ depends on $x(i)$
  - set of samples $x^{(1)}, x^{(2)}, \ldots$ used to approximate $p(x)$



$$\mathbf{x}^{(1)} = (X_1 = x_1^{(1)}, X_2 = x_2^{(1)}, \ldots, X_5 = x_5^{(1)})$$

$$\mathbf{x}^{(2)} = (X_1 = x_1^{(2)}, X_2 = x_2^{(2)}, \ldots, X_5 = x_5^{(2)})$$

$$\mathbf{x}^{(3)} = (X_1 = x_1^{(3)}, X_2 = x_2^{(3)}, \ldots, X_5 = x_5^{(3)})$$

Slide by Domingos

Markov Chain Monte Carlo

- **Gibbs Sampler**
  1. Start with an initial assignment to nodes
  2. One node at a time, sample node given others
  3. Repeat
  4. Use samples to compute P(X)
- **Convergence: Burn-in + Mixing time**
- **Many modes $\Rightarrow$ Multiple chains**

Iterations required to move away from particular initial condition

Iterations required to be close to stationary dist.

Slide by Domingos

Other Inference Methods

- Belief propagation (sum-product)
- Mean field / Variational approximations

Slide by Domingos

Learning

- **Learning Weights**
  - Maximize likelihood
  - Convex optimization: gradient ascent, quasi-Newton methods, etc.
  - Requires inference at each step (slow!)

- **Learning Structure**
  - Feature Search
  - Evaluation using Likelihood, …

Back to CRFs

- **CRFs are conditionally trained Markov Networks**

Linear-Chain Conditional Random Fields

- **From HMMs to CRFs**

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

can also be written as

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

(set $\lambda_{ij} := \log p(y' = i | y = j)$, ...)

We let new parameters vary freely, so we need normalization constant Z.

Linear-Chain Conditional Random Fields

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

- **Introduce feature functions** $f_k(y_t, y_{t-1}, x_t\ldots$

One feature per transition

One feature per state-observati…

$$(f_{ij}(y,y',x_t) := \mathbf{1}_{y=i}\mathbf{1}_{y'=j} \quad , \quad f_{io}(y,y',x_t) := \mathbf{1}_{y=i}\mathbf{1}_{x=o}$$

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

- **Then the conditional distribution is**

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

This is a linear-chain CRF, but includes only current word's identity as a feature

Linear-Chain Conditional Random Fields

- Conditional $p(y|x)$
  that follows from joint $p(y,x)$ of HMM
  is a linear CRF with certain feature
  functions!

Linear-Chain Conditional Random Fields

- **Definition:**
  A *linear-chain CRF* is a distribution that takes the form

parameters

feature functions

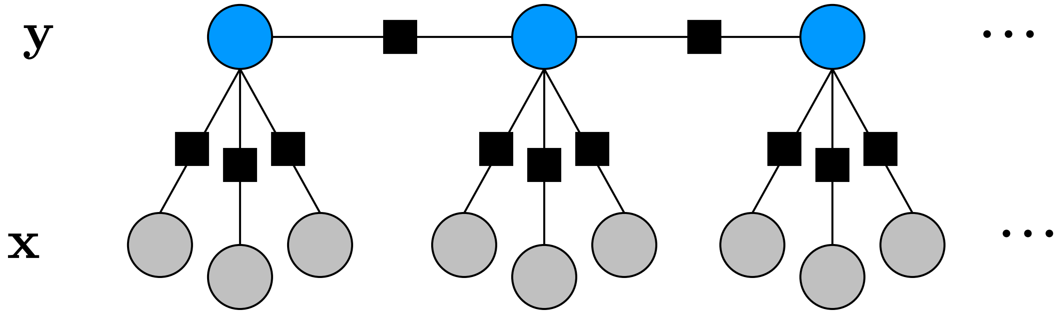
$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

where Z(x) is a normalization function

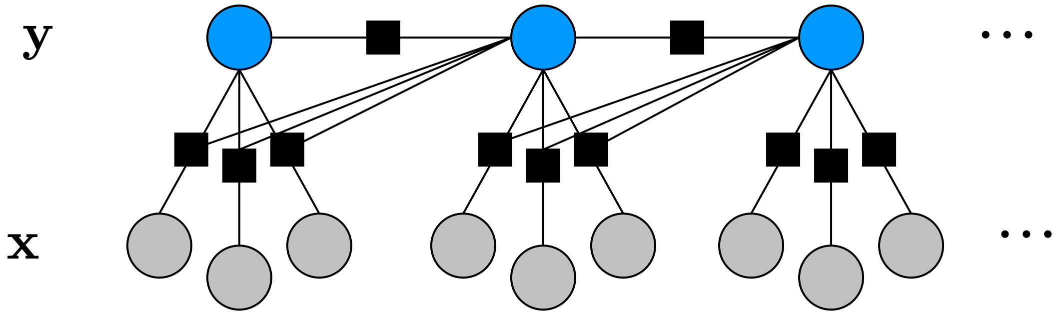
$$Z(x) = \sum_{\mathbf{y}} exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

Linear-Chain Conditional Random Fields

- HMM-like linear-chain CRF



- Linear-chain CRF, in which transition score depends on the current observation



Questions

- **#1 – Inference**

  Given observations $x_1 \ldots x_N$ and CRF θ, what is $P(y_t, y_{t-1}|x)$ and what is $Z(x)$? (needed for learning)

- **#2 – Inference**

  Given observations $x_1 \ldots x_N$ and CRF θ, what is the most likely (Viterbi) labeling $y^* = \arg\max_y p(y|x)$?

- **#3 – Learning**

  Given iid training data $D=\{x^{(i)}, y^{(i)}\}, i=1..N$, how do we estimate the parameters $\theta=\{\lambda_k\}$ of a linear-chain CRF?

Solutions to #1 and #2

- **Forward/Backward and Viterbi algorithms similar to versions for HMMs**
- **HMM as factor graph**

HMM Definition

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

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

$$\mathbf{\Psi}_t(j,i,x) := p(y_t = j|y_{t-1} = i)p(x_t = x|y_t = j)$$

- **Then**

$$\alpha_t(i) = \sum_{i \in S} \mathbf{\Psi}_t(j,i,x_t)\alpha_{t-1}(i)$$ forward recursion

$$\beta_t(i) = \sum_{j \in S} \mathbf{\Psi}_{t+1}(j,i,x_{t+1})\beta_{t+1}(j)$$ backward recursion

$$\delta_t(j) = \max_{i \in S} \mathbf{\Psi}_t(j,i,x_t)\delta_{t-1}(i)$$ Viterbi recursion

Forward/Backward for linear-chain CRFs …

- … identical to HMM version except for factor functions $\mathbf{\Psi}_t(j, i, \mathbf{x}_t)$
- CRF can be written as

CRF Definition

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

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{t=1}^{T} \mathbf{\Psi}_t(y_t, y_{t-1}, \mathbf{x}_t)$$

$$\mathbf{\Psi}_t(y_t, y_{t-1}, \mathbf{x}_t) := exp\left(\sum_{k} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

- Same:

$$\alpha_t(i) = \sum_{i \in S} \mathbf{\Psi}_t(j, i, x_t)\alpha_{t-1}(i)$$ forward recursion

$$\beta_t(i) = \sum_{j \in S} \mathbf{\Psi}_{t+1}(j, i, x_{t+1})\beta_{t+1}(j)$$ backward recursion

$$\delta_t(j) = \max_{i \in S} \mathbf{\Psi}_t(j, i, x_t)\delta_{t-1}(i)$$ Viterbi recursion

Forward/Backward for linear-chain CRFs

- **Complexity same as for HMMs**

<u>Time:</u>

$O(K^2N)$

Linear in length of sequence!

<u>Space:</u>

$O(KN)$

K = |S| #states

N length of sequence

Solution to #3 - Learning

- **Want to maximize Conditional log likelihood**

$$l(\theta) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)})$$

- **Substitute in CRF model into likelihood**

CRFs typically learned using numerical

$$l(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)})$$

$$- \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

(Also possible for HMMs, but we only discussed EM)

- **Add Regularizer**

Often large number of parameters, so need to avoid overfitting

Regularization

- **Commonly used l_2-norm (Euclidean)**
  - Corresponds to Gaussian prior over parameters

$$-\sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

- **Alternative is l_1-norm**
  - Corresponds to exponential prior over parameters
  - Encourages sparsity

$$-\sum_{k=1}^{K} \frac{|\lambda_k|}{\sigma}$$

- **Accuracy of final model not sensitive to σ**

Optimizing the Likelihood

- There exists no closed-form solution, so must use numerical optimization.

$$l(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

$$\frac{\partial l}{\partial \lambda_k} = \sum_{i=1}^{N} \sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y,y'} f_k(y, y', \mathbf{x}_t^{(i)}) p(y, y' | \mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2}$$

- $l(\theta)$ is concave and with regularizer strictly concave

  → only one global optimum

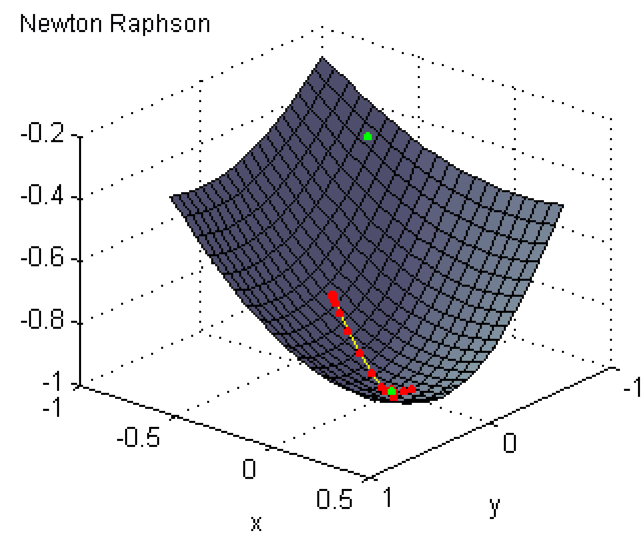


Figure by Cohen & McCallum

Optimizing the Likelihood

- **Steepest Ascent**

  very slow!

- **Newton's method**

  fewer iterations, but requires Hessian^{-1}

- **Quasi-Newton methods**

  approximate Hessian by analyzing successive gradients

  - BFGS

    fast, but approximate Hessian requires quadratic space

  - L-BFGS (limited-memory)

    fast even with limited memory!

  - Conjugate Gradient

Computational Cost

$$l(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log \boxed{Z(\mathbf{x}^{(i)})} - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

$$\frac{\partial l}{\partial \lambda_k} = \sum_{i=1}^{N}\sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k(y, y', \mathbf{x}_t^{(i)}) \boxed{p(y, y' | \mathbf{x}^{(i)})} - \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2}$$

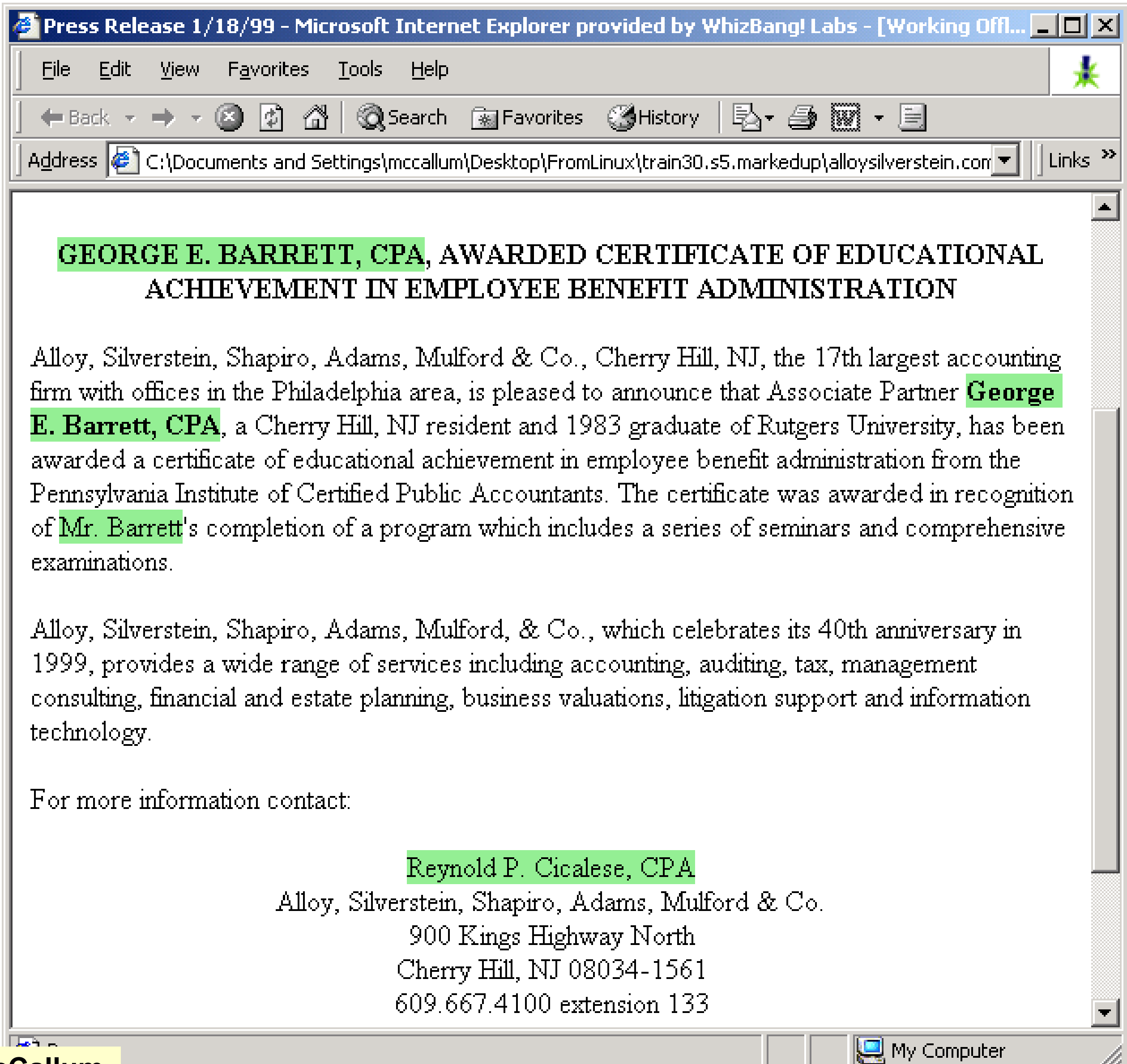
- **For each training instance:**
  O(K²T) **(using forward-backward)**
- **For N training instances, G iterations:**
  O(K²TNG)

Examples:

- Named-entity recognition   11 labels; 200,000 words   < 2 hours
- Part-of-speech tagging   45 labels, 1 million words   > 1 week

Person name Extraction

[McCallum 2001, unpublished]


Press Release 1/18/99 - Microsoft Internet Explorer provided by WhizBang! Labs - [Working Offl...

File Edit View Favorites Tools Help

Address C:\Documents and Settings\mccallum\Desktop\FromLinux\train30.s5.markedup\alloysilverstein.com

GEORGE E. BARRETT, CPA, AWARDED CERTIFICATE OF EDUCATIONAL ACHIEVEMENT IN EMPLOYEE BENEFIT ADMINISTRATION

Alloy, Silverstein, Shapiro, Adams, Mulford & Co., Cherry Hill, NJ, the 17th largest accounting firm with offices in the Philadelphia area, is pleased to announce that Associate Partner **George E. Barrett, CPA**, a Cherry Hill, NJ resident and 1983 graduate of Rutgers University, has been awarded a certificate of educational achievement in employee benefit administration from the Pennsylvania Institute of Certified Public Accountants. The certificate was awarded in recognition of Mr. Barrett's completion of a program which includes a series of seminars and comprehensive examinations.

Alloy, Silverstein, Shapiro, Adams, Mulford, & Co., which celebrates its 40th anniversary in 1999, provides a wide range of services including accounting, auditing, tax, management consulting, financial and estate planning, business valuations, litigation support and information technology.

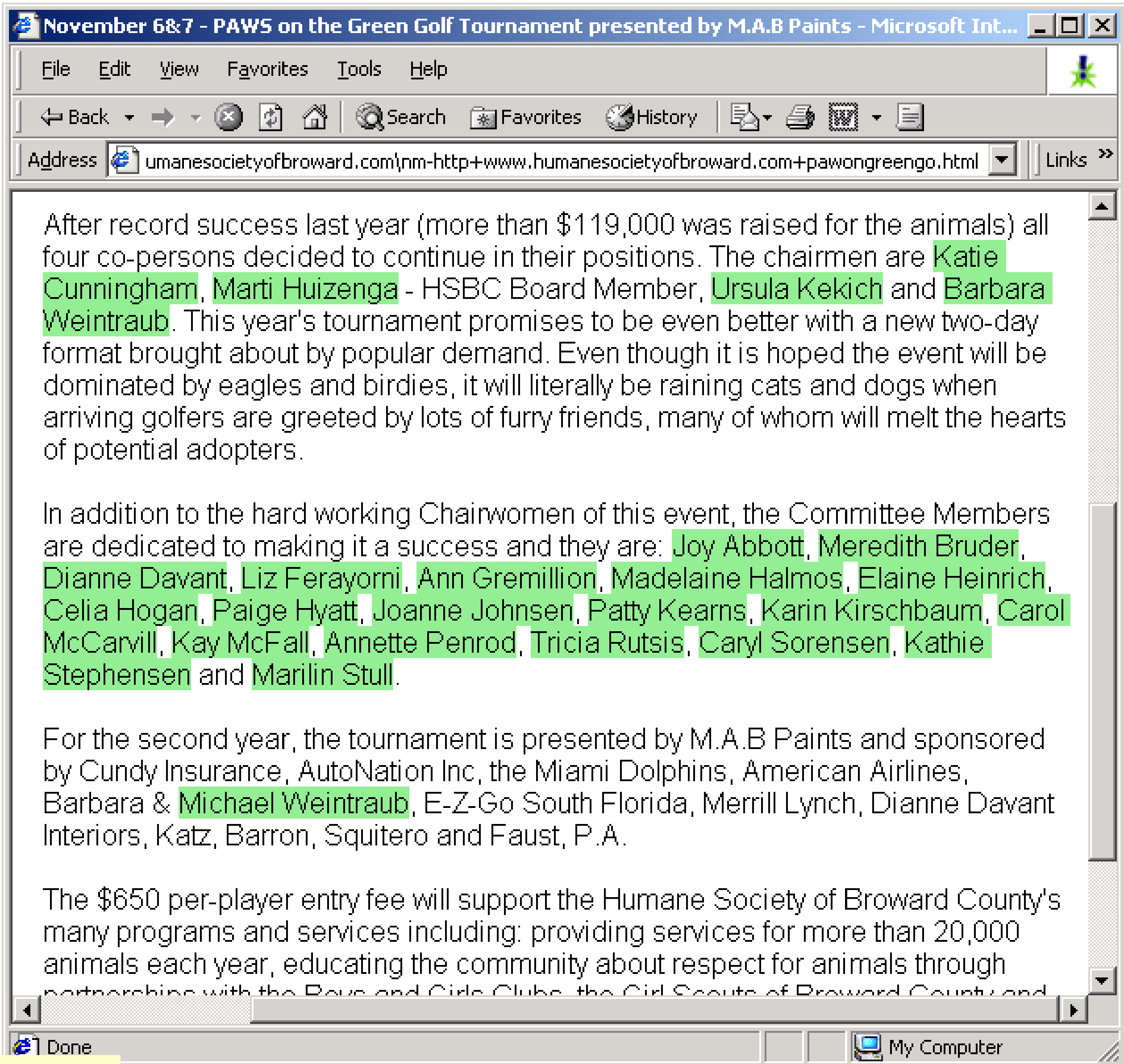
For more information contact:

Reynold P. Cicalese, CPA
Alloy, Silverstein, Shapiro, Adams, Mulford & Co.
900 Kings Highway North
Cherry Hill, NJ 08034-1561
609.667.4100 extension 133

My Computer

Slide by Cohen & McCallum

Person name Extraction



After record success last year (more than $119,000 was raised for the animals) all four co-persons decided to continue in their positions. The chairmen are Katie Cunningham, Marti Huizenga - HSBC Board Member, Ursula Kekich and Barbara Weintraub. This year's tournament promises to be even better with a new two-day format brought about by popular demand. Even though it is hoped the event will be dominated by eagles and birdies, it will literally be raining cats and dogs when arriving golfers are greeted by lots of furry friends, many of whom will melt the hearts of potential adopters.

In addition to the hard working Chairwomen of this event, the Committee Members are dedicated to making it a success and they are: Joy Abbott, Meredith Bruder, Dianne Davant, Liz Ferayorni, Ann Gremillion, Madelaine Halmos, Elaine Heinrich, Celia Hogan, Paige Hyatt, Joanne Johnsen, Patty Kearns, Karin Kirschbaum, Carol McCarvill, Kay McFall, Annette Penrod, Tricia Rutsis, Caryl Sorensen, Kathie Stephensen and Marilin Stull.

For the second year, the tournament is presented by M.A.B Paints and sponsored by Cundy Insurance, AutoNation Inc, the Miami Dolphins, American Airlines, Barbara & Michael Weintraub, E-Z-Go South Florida, Merrill Lynch, Dianne Davant Interiors, Katz, Barron, Squitero and Faust, P.A.

The $650 per-player entry fee will support the Humane Society of Broward County's many programs and services including: providing services for more than 20,000 animals each year, educating the community about respect for animals through

Slide by Cohen & McCallum

Features in Experiment

Capitalized	Xxxxx
Mixed Caps	XxXxxx
All Caps	XXXXX
Initial Cap	X….
Contains Digit	xxx5
All lowercase	xxxx
Initial	X
Punctuation	.,:;!(), etc
Period	.
Comma	,
Apostrophe	'
Dash	-
Preceded by HTML tag	

Character n-gram classifier says string is a person name (80% accurate)

In stopword list (the, of, their, etc)

In honorific list (Mr, Mrs, Dr, Sen, etc)

In person suffix list (Jr, Sr, PhD, etc)

In name particle list (de, la, van, der, etc)

In Census lastname list; segmented by P(name)

In Census firstname list; segmented by P(name)

In locations lists (states, cities, countries)

In company name list ("J. C. Penny")

In list of company suffixes (Inc, & Associates, Foundation)

Hand-built FSM person-name extractor says yes, (prec/recall ~ 30/95)

Conjunctions of all previous feature pairs, evaluated at the current time step.

Conjunctions of all previous feature pairs, evaluated at current step and one step ahead.

All previous features, evaluated two steps ahead.

All previous features, evaluated one step behind.

Total number of features = ~500k

Slide by Cohen & McCallum

Training and Testing

- Trained on 65k words from 85 pages, 30 different companies' web sites.
- Training takes 4 hours on a 1 GHz Pentium.
- Training precision/recall is 96% / 96%.

- Tested on different set of web pages with similar size characteristics.
- Testing precision is 92 – 95%,
  recall is 89 – 91%.

Slide by Cohen & McCallum

Part-of-speech Tagging

45 tags, 1M words training data, Penn Treebank

DT NN NN , NN , VBZ RB JJ IN
The asbestos fiber , crocidolite, is unusually resilient once

PRP VBZ DT NNS , IN RB JJ NNS TO PRP VBG
it enters the lungs , with even brief exposures to it causing

NNS WDT VBP RP NNS JJ , NNS VBD .
symptoms that show up decades later , researchers said .

			Using spelling features*			
	Error	oov error	error	Δ err	oov error	Δ err
HMM	5.69%	45.99%				
CRF	5.55%	48.05%	4.27%	-24%	23.76%	-50%

* use words, *plus* overlapping features: capitalized, begins with #, contains hyphen, ends in -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies.

[Lafferty, McCallum, Pereira 2001]

Slide by Cohen & McCallum

Table Extraction from Government Reports

Cash receipts from marketings of milk during 1995 at $19.9 billion dollars, was slightly below 1994. Producer returns averaged $12.93 per hundredweight, $0.19 per hundredweight below 1994. Marketings totaled 154 billion pounds, 1 percent above 1994. Marketings include whole milk sold to plants and dealers as well as milk sold directly to consumers.

An estimated 1.56 billion pounds of milk were used on farms where produced, 8 percent less than 1994. Calves were fed 78 percent of this milk with the remainder consumed in producer households.

Milk Cows and Production of Milk and Milkfat:
United States, 1993-95

Year	Number of Milk Cows 1/	Production of Milk and Milkfat 2/				
		Per Milk Cow		Percentage of Fat in All Milk Produced	Total	
		Milk	Milkfat		Milk	Milkfat
	1,000 Head	--- Pounds ---		Percent	Million Pounds	
1993	9,589	15,704	575	3.66	150,582	5,514.4
1994	9,500	16,175	592	3.66	153,664	5,623.7
1995	9,461	16,451	602	3.66	155,644	5,694.3

1/ Average number during year, excluding heifers not yet fresh.
2/ Excludes milk sucked by calves.

Table Extraction from Government Reports

[Pinto, McCallum, Wei, Croft, 2003]

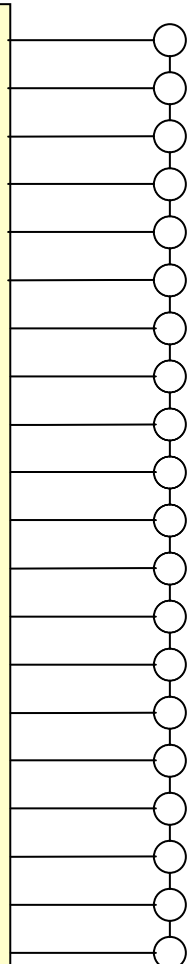
100+ documents from www.fedstats.gov

```
of milk during 1995 at $19.9 billion dollars, was
eturns averaged $12.93 per hundredweight,
 1994.  Marketings totaled 154 billion pounds,
gs include whole milk sold to plants and dealers
consumers.

ls of milk were used on farms where produced,
s were fed 78 percent of this milk with the
er households.


ction of Milk and Milkfat:
1993-95
------------------------------------
n of Milk and Milkfat 2/
---------------------------------------
w    :  Percentage  :      Total
----: of Fat in All  :------------------
 Milk Produced  : Milk  : Milkfat
------------------------------------
```

CRF



Labels:

- Non-Table
- Table Title
- Table Header
- Table Data Row
- Table Section Data Row
- Table Footnote
- ... *(12 in all)*

Features:

- Percentage of digit chars
- Percentage of alpha chars
- Indented
- Contains 5+ consecutive spaces
- Whitespace in this line aligns with prev.
- ...
- Conjunctions of all previous features, time offset: {0,0}, {-1,0}, {0,1}, {1,2}.

Slide by Cohen & McCallum

Table Extraction Experimental Results

[Pinto, McCallum, Wei, Croft, 2003]

	Line labels, percent correct
HMM	65 %
Stateless MaxEnt	85 %
CRF w/out conjunctions	52 %
CRF	95 %

Δ error = 85%

Slide by Cohen & McCallum

Named Entity Recognition

Reuters stories on international news

Train on ~300k words

CRICKET -
MILLNS SIGNS FOR BOLAND

CAPE TOWN 1996-08-22

South African provincial side Boland said on Thursday they had signed Leicestershire fast bowler David Millns on a one year contract.
Millns, who toured Australia with England A in 1992, replaces former England all-rounder Phillip DeFreitas as Boland's overseas professional.

Labels:	Examples:
PER	Yayuk Basuki Innocent Butare
ORG	3M KDP Leicestershire
LOC	Leicestershire Nirmal Hriday The Oval
MISC	Java Basque 1,000 Lakes Rally

Slide by Cohen & McCallum

Automatically Induced Features

[McCallum 2003]

Index	*Feature*
0	inside-noun-phrase (o_{t-1})
5	stopword (o_t)
20	capitalized (o_{t+1})
75	word=the (o_t)
100	in-person-lexicon (o_{t-1})
200	word=in (o_{t+2})
500	word=Republic (o_{t+1})
711	word=RBI (o_t) & header=BASEBALL
1027	header=CRICKET (o_t) & in-English-county-lexicon (o_t)
1298	company-suffix-word ($\text{firstmention}_{t+2}$)
4040	location (o_t) & POS=NNP (o_t) & capitalized (o_t) & stopword (o_{t-1})
4945	moderately-rare-first-name (o_{t-1}) & very-common-last-name (o_t)
4474	word=the (o_{t-2}) & word=of (o_t)

Slide by Cohen & McCallum

Named Entity Extraction Results

[McCallum & Li, 2003]

Method	F1	# parameters
BBN's Identinfinder, word features	79%	~500k
CRFs word features, w/out Feature Induction	80%	~500k
CRFs many features, w/out Feature Induction	75%	~3 million
CRFs many candidate features with Feature Induction	90%	~60k

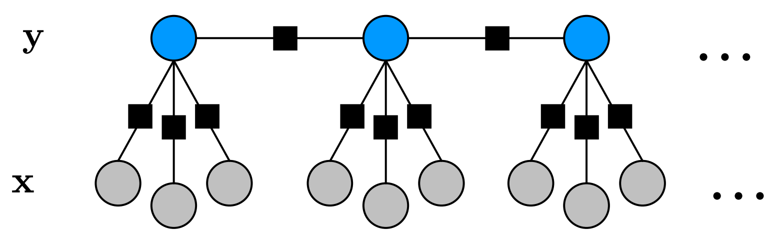
So far ...

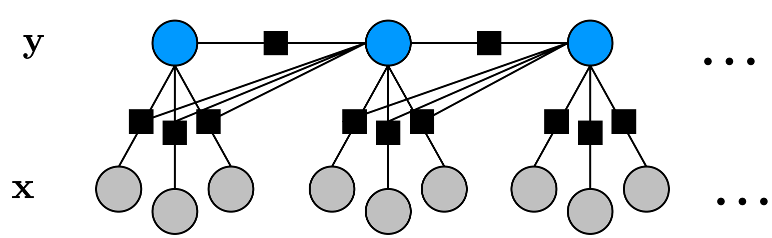
- ... only looked at linear-chain CRFs

parameters

feature functions

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





General CRFs vs. HMMs

- More general and expressive modeling technique
- Comparable computational efficiency
- Features may be arbitrary functions of *any* or *all* observations
- Parameters need not fully specify generation of observations; require less training data
- Easy to incorporate domain knowledge
- State means only "state of process", vs "state of process" and "observational history I'm keeping"

Slide by Cohen & McCallum

General CRFs

- **Definition**
  - Let G be a factor graph. Then p(y|x) is a CRF if for any x, p(y|x) factorizes according to G.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_A \in G} \exp\left(\sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\mathbf{y}_A, \mathbf{x}_A)\right)$$

But often some parameters tied:
Clique Templates

For compar

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

Questions

- **#1 – Inference**

  Again, learning requires computing $P(y_c|x)$ for given observations $x_1 \dots x_N$ and CRF θ.

- **#2 – Inference**

  Given observations $x_1 \dots x_N$ and CRF θ, what is the most likely labeling $y^* = \arg\max_y p(y|x)$?

- **#3 – Learning**

  Given iid training data $D=\{x^{(i)}, y^{(i)}\}, i=1..N$, how do we estimate the parameters $\theta=\{ \lambda_k \}$ of a CRF?

Inference

- **For graphs with small treewidth**
  - Junction Tree Algorithm
- **Otherwise approximate inference**
  - Sampling-based approaches:
    MCMC, …
    - Not useful for training (too slow for every iteration)
  - Variational approaches:
    Belief Propagation, …
    - Popular

Learning

- **Similar to linear-chain case**
- **Substitute model into likelihood …**

inference

$$l(\theta) = \sum_{C_p \in \mathcal{C}} \sum_{\Psi_c \in C_p} \sum_{k=1}^{K(p)} \lambda_{pk} f_{pk}(\mathbf{x}_x, \mathbf{y}_c) - \log \boxed{Z(\mathbf{x})}$$

… and compute partial derivatives, …

$$\frac{\partial l}{\partial \lambda_{pk}} = \sum_{\Psi_c \in C_p} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) - \sum_{\Psi_c \in C_p} \sum_{\mathbf{y}'_c} f_{pk}(\mathbf{x}_c, \mathbf{y}'_c) \boxed{p(\mathbf{y}'_c|\mathbf{x})}$$

and run nonlinear optimization (L-BFGS)

Markov Logic

- **A general language capturing logic and uncertainty**
- **A Markov Logic Network (MLN) is a set of pairs (F, w) where**
  - F is a formula in first-order logic
  - w is a real number
- **Together with constants, it defines a Markov network with**
  - One node for each ground predicate
  - One feature for each ground formula F, with the corresponding weight w

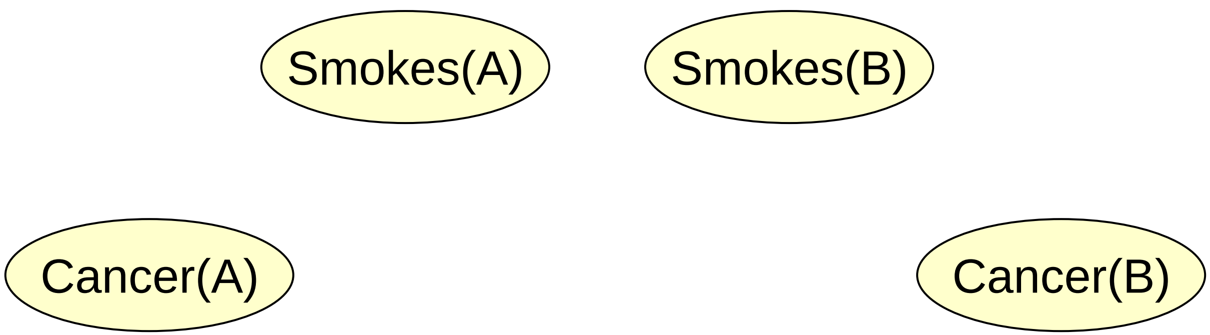
$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x) \right)$$

Slide by Poon

Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)

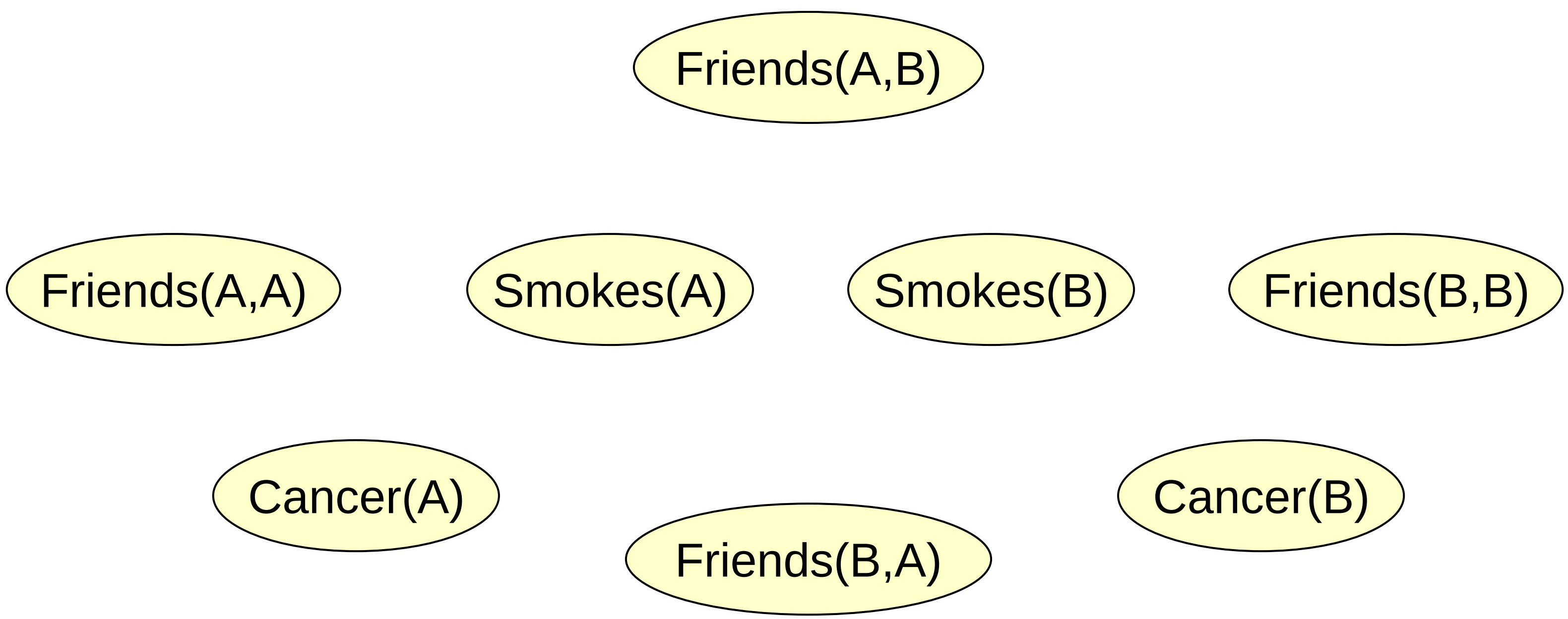


Slide by Domingos

Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

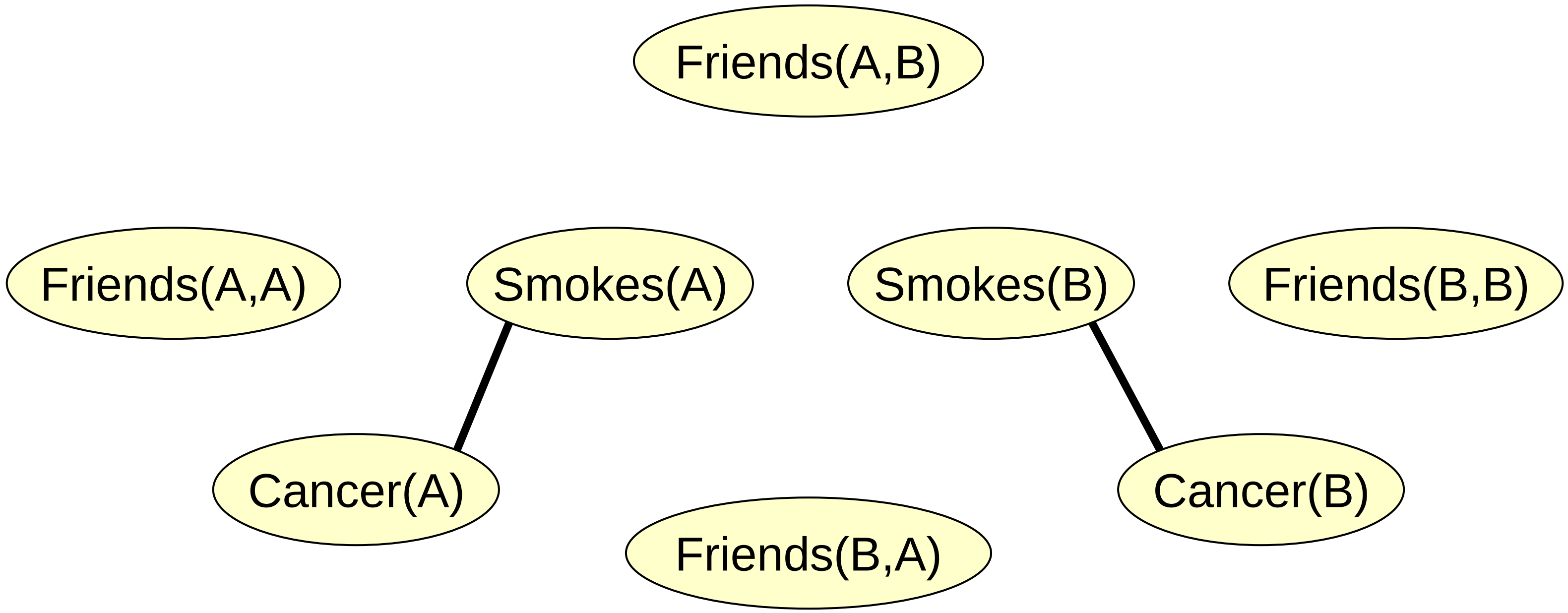
Suppose we have two constants: **Anna** (A) and **Bob** (B)



Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

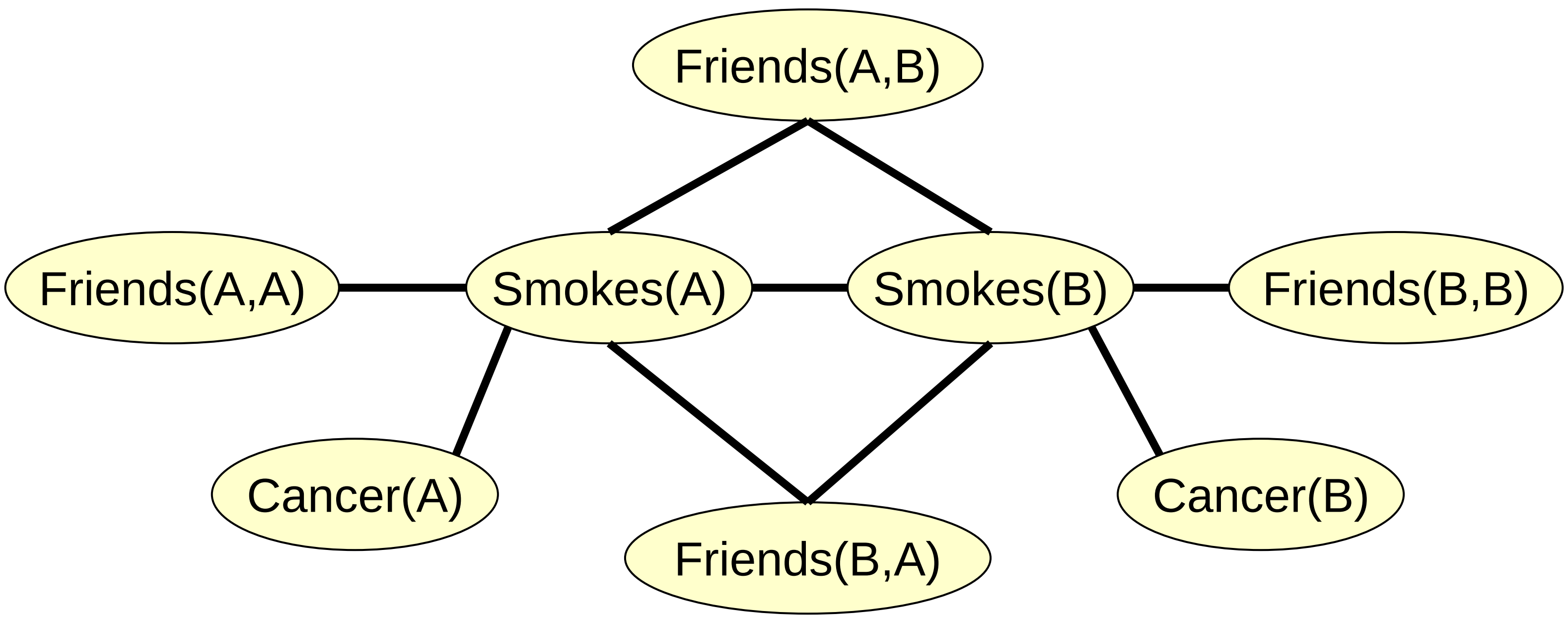
Suppose we have two constants: **Anna** (A) and **Bob** (B)



Example of an MLN

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



Slide by Domingos

Joint Inference in Information Extraction

Hoifung Poon

Dept. Computer Science & Eng.
University of Washington

(Joint work with Pedro Domingos)

Problems of Pipeline Inference

- **AI systems typically use pipeline architecture**
  - Inference is carried out in stages
  - E.g., information extraction, natural language processing, speech recognition, vision, robotics
- **Easy to assemble & low computational cost, but …**
  - Errors accumulate along the pipeline
  - No feedback from later stages to earlier ones
- **Worse: Often process one object at a time**

Slide by Poon

We Need Joint Inference

?

Author Title

S. Minton Integrating heuristics for constraint satisfaction problems: A case study. In AAAI Proceedings, 1993.

Minton, S(1993 b). Integrating heuristics for constraint satisfaction problems: A case study. In: Proceedings AAAI.