Learning Structural SVMs with Latent Variables

Chun-Nam Yu & Thorsten Joachims

Dept. of Computer Science, Cornell University

June 17, ICML '09

Cornell University
Computer Science

Latent Variable Models

- Widely used in statistics and machine learning

Latent Variable Models

- Widely used in statistics and machine learning
- Unobserved quantities/missing data in experiments

Latent Variable Models

- Widely used in statistics and machine learning
- Unobserved quantities/missing data in experiments
- Dimensionality reduction

Latent Variable Models

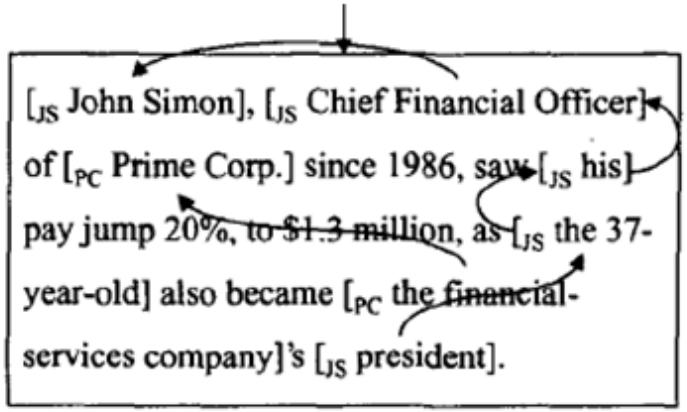
- Widely used in statistics and machine learning
- Unobserved quantities/missing data in experiments
- Dimensionality reduction
- Classical examples: Factor Analysis, Mixture Models, PCA

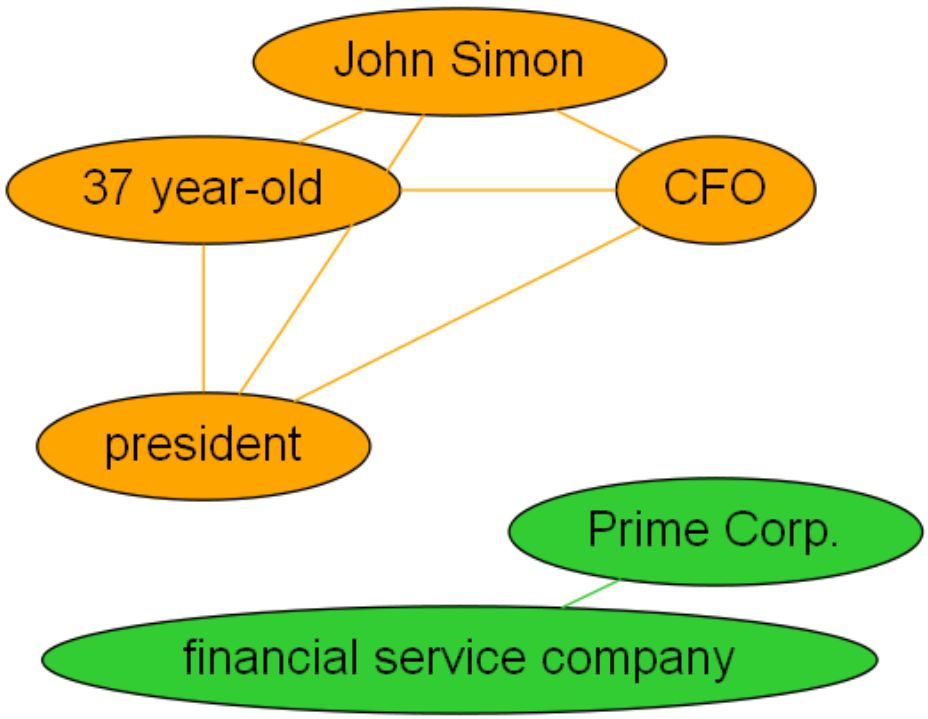
Latent Variable Models

- Widely used in statistics and machine learning
- Unobserved quantities/missing data in experiments
- Dimensionality reduction
- Classical examples: Factor Analysis, Mixture Models, PCA
- This talk: Latent variables in *supervised prediction tasks*

An Example: Noun Phrase Coreference

[from Cardie & Wagstaff '99]

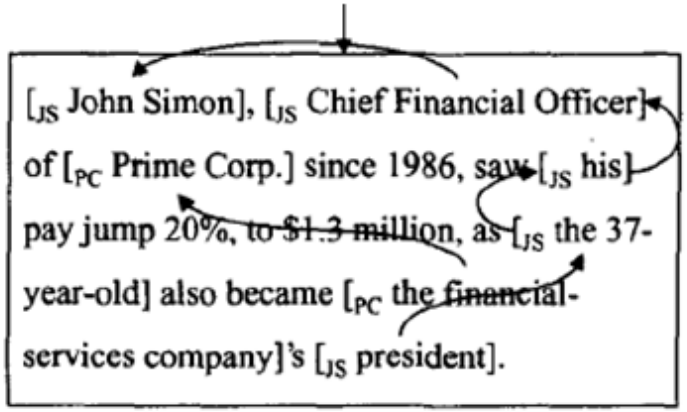


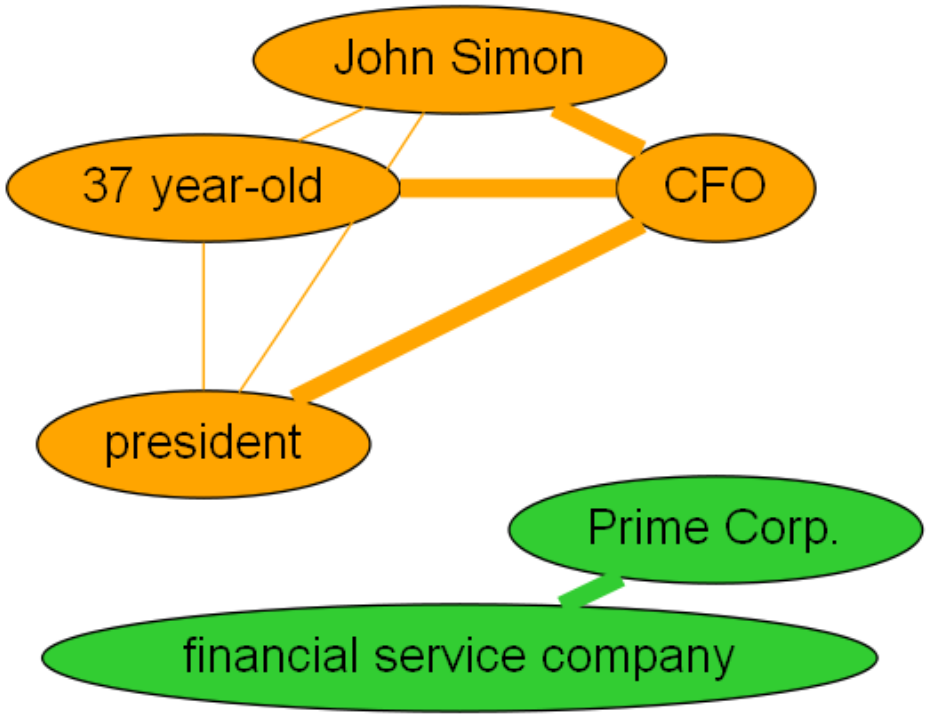


- Document with many noun phrases, want to determine which refer to the same entity

An Example: Noun Phrase Coreference

[from Cardie & Wagstaff '99]

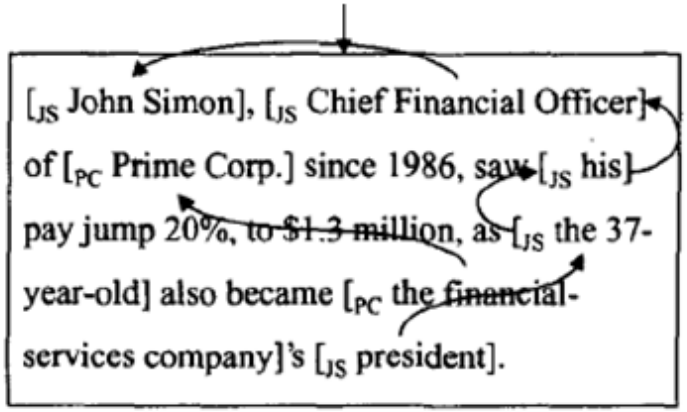


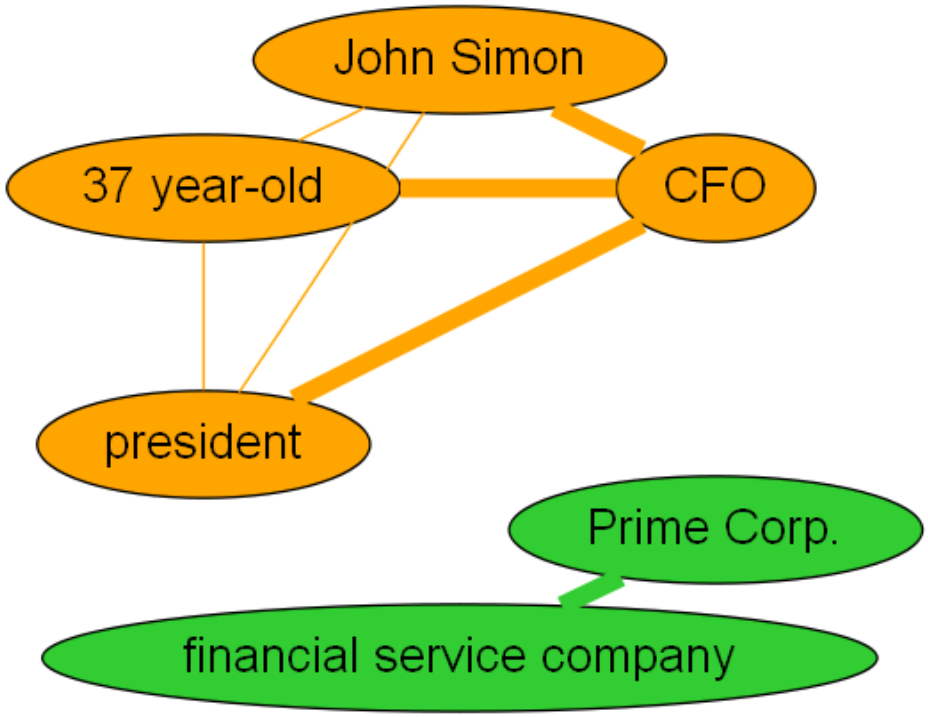


- Document with many noun phrases, want to determine which refer to the same entity
- Not all links are informative. Humans follow 'strong' links and reason transitively for coreference.

An Example: Noun Phrase Coreference

[from Cardie & Wagstaff '99]





- Document with many noun phrases, want to determine which refer to the same entity
- Not all links are informative. Humans follow 'strong' links and reason transitively for coreference.
- Other examples: Machine Translation

Latent Variables in Structured Output Learning

- Generative Models
  - Hidden Markov Models in speech recognition and bioinformatics

Latent Variables in Structured Output Learning

- Generative Models
  - Hidden Markov Models in speech recognition and bioinformatics
- Discriminative Probabilistic Models
  - Hidden Conditional Random Fields [Wang et al. '06]
  - Discriminative PCFG with latent annotations [Petrov & Klein '07]

Latent Variables in Structured Output Learning

- Generative Models
  - Hidden Markov Models in speech recognition and bioinformatics
- Discriminative Probabilistic Models
  - Hidden Conditional Random Fields [Wang et al. '06]
  - Discriminative PCFG with latent annotations [Petrov & Klein '07]
- Large-Margin Models
  - Large-margin criterion for Hidden CRF in vision community [Felzenswalb et al. '08, Wang & Mohri '08]
  - Semi-supervised Structural SVM [Zien & Brefeld '07]

Structural SVMs

- Want to learn a linear prediction rule:

$$f_{\vec{w}}(x) = \text{argmax}_{y \in \mathcal{Y}} \, \vec{w} \cdot \Phi(x, y)$$

Structural SVM (Margin rescaling)

$$\min_{\vec{w}, \vec{\xi}} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$s.t. \text{ for } 1 \leq i \leq n, \text{ for all output structures } \hat{y} \in \mathcal{Y},$$

$$\vec{w} \cdot \Phi(x_i, y_i) - \vec{w} \cdot \Phi(x_i, \hat{y}) \geq \Delta(y_i, \hat{y}) - \xi_i$$

Structural SVMs

- Want to learn a linear prediction rule:

$$f_{\vec{w}}(x) = \text{argmax}_{y \in \mathcal{Y}} \, \vec{w} \cdot \Phi(x, y)$$

Structural SVM (Margin rescaling)

$$\min_{\vec{w}, \vec{\xi}} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$s.t. \text{ for } 1 \leq i \leq n, \text{ for all output structures } \hat{y} \in \mathcal{Y},$$

$$\vec{w} \cdot \Phi(x_i, y_i) - \vec{w} \cdot \Phi(x_i, \hat{y}) \geq \Delta(y_i, \hat{y}) - \xi_i$$

- Loss function Δ controls margin between correct label y_i and incorrect predictions $\hat{y}$

Structural SVMs

- Want to learn a linear prediction rule:

$$f_{\vec{w}}(x) = \text{argmax}_{y \in \mathcal{Y}} \, \vec{w} \cdot \Phi(x, y)$$

Structural SVM (Margin rescaling)

$$\min_{\vec{w}, \vec{\xi}} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$s.t. \text{ for } 1 \leq i \leq n, \text{ for all output structures } \hat{y} \in \mathcal{Y},$$

$$\vec{w} \cdot \Phi(x_i, y_i) - \vec{w} \cdot \Phi(x_i, \hat{y}) \geq \Delta(y_i, \hat{y}) - \xi_i$$

- Loss function Δ controls margin between correct label y_i and incorrect predictions $\hat{y}$
- Great prediction performance on many structured prediction problems

Introducing Latent Variables to Structural SVMs

- How can we extend structural SVM to handle latent variables?

Introducing Latent Variables to Structural SVMs

- How can we extend structural SVM to handle latent variables?
- Many interesting question to consider:
  - Feature representation and loss function?

Introducing Latent Variables to Structural SVMs

- How can we extend structural SVM to handle latent variables?
- Many interesting question to consider:
  - Feature representation and loss function?
  - Non-convex training objective?

Introducing Latent Variables to Structural SVMs

- How can we extend structural SVM to handle latent variables?
- Many interesting question to consider:
  - Feature representation and loss function?
  - Non-convex training objective?
  - Inference problems?

Latent Structural SVMs - Prediction Rules

- Extends the joint feature map $\Phi(x, y)$ to $\Phi(x, y, h)$.

Latent Structural SVMs - Prediction Rules

- Extends the joint feature map $\Phi(x, y)$ to $\Phi(x, y, h)$.
- A new argmax prediction rule:

$$f_{\vec{w}}(x) = (\bar{y}, \bar{h}) = \text{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}} \, \vec{w} \cdot \Phi(x, y, h)$$

Latent Structural SVMs - Prediction Rules

- Extends the joint feature map $\Phi(x, y)$ to $\Phi(x, y, h)$.
- A new argmax prediction rule:

$$f_{\vec{w}}(x) = (\bar{y}, \bar{h}) = \text{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}} \vec{w} \cdot \Phi(x, y, h)$$

- Linear scoring rule; requires joint inference over y and h

Latent Structural SVMs - Formulation

Optimization Problem for Latent Structural SVM

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Assumes Δ does not depend on the latent variable h

Latent Structural SVMs - Formulation

Optimization Problem for Latent Structural SVM

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Assumes Δ does not depend on the latent variable h
- ξ_i upper bounds the loss

Latent Structural SVMs - Formulation

Optimization Problem for Latent Structural SVM

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Assumes Δ does not depend on the latent variable h
- ξ_i upper bounds the loss
- Reduces to standard structural SVM if latent variable not present

Latent Structural SVMs - Formulation

- In Structural SVM prediction loss is bounded by:

$$\Delta(y_i, f_{\vec{w}}(x_i)) \leq \underbrace{\max_{\hat{y} \in \mathcal{Y}} [\vec{w} \cdot \Phi(x_i, \hat{y}) + \Delta(y_i, \hat{y})]}_{convex} - \underbrace{\vec{w} \cdot \Phi(x_i, y_i)}_{linear} = \xi_i$$

Latent Structural SVMs - Formulation

- In Structural SVM prediction loss is bounded by:

$$\Delta(y_i, f_{\vec{w}}(x_i)) \leq \underbrace{\max_{\hat{y}\in\mathcal{Y}}[\vec{w}\cdot\Phi(x_i,\hat{y}) + \Delta(y_i,\hat{y})]}_{convex} - \underbrace{\vec{w}\cdot\Phi(x_i,y_i)}_{linear} = \xi_i$$

- In the case with latent variables:

$$\begin{aligned}&\Delta(y_i, f_{\vec{w}}(x_i))\\ \leq &\underbrace{\max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y},\hat{h})]}_{convex} \underbrace{- \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)}_{concave} = \xi_i\end{aligned}$$

Latent Structural SVMs - Formulation

- Why have the restriction on loss function Δ?

$$\max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta((y_i,h),(\hat{y},\hat{h}))] - \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)$$

Latent Structural SVMs - Formulation

- Why have the restriction on loss function Δ?

$$\max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta((y_i,h),(\hat{y},\hat{h}))] - \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)$$

- Inclusion of h introduce dependencies between convex term and concave term

Latent Structural SVMs - Formulation

- Why have the restriction on loss function Δ?

$$\max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta((y_i,h),(\hat{y},\hat{h}))] - \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)$$

- Inclusion of h introduce dependencies between convex term and concave term
- Allows efficient solution algorithm while retaining modeling power

Non-Convex Objective - The CCCP Algorithm

- Concave-Convex Procedure[Yuille & Rangarajan '03]
- Employed in many recent machine learning work [Collobert et al. '06, Smola et al. '05, Chapelle et al. '08]

Non-Convex Objective - The CCCP Algorithm

Concave-Convex Procedure (CCCP)

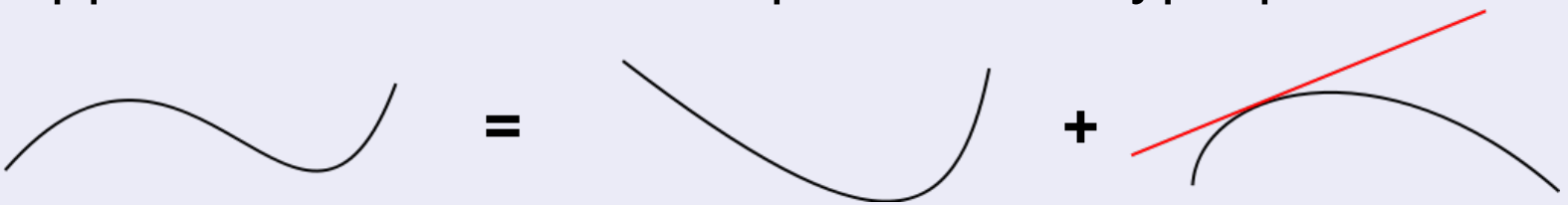
1. Decompose the objective into convex and concave part

Non-Convex Objective - The CCCP Algorithm

Concave-Convex Procedure (CCCP)

1. Decompose the objective into convex and concave part

2. Upper bound the concave part with a hyperplane

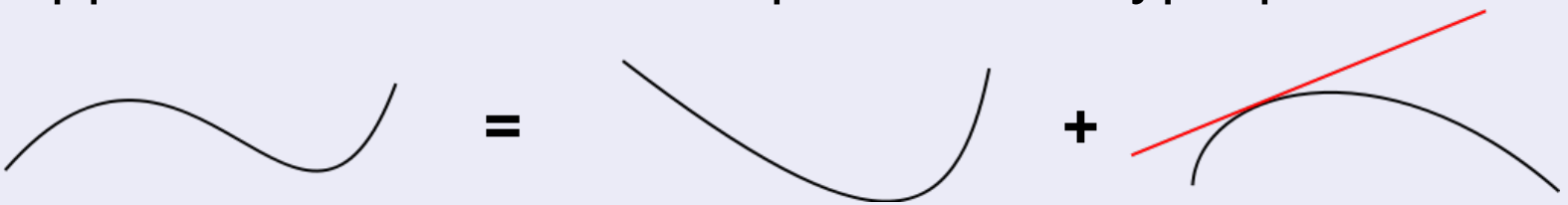


Non-Convex Objective - The CCCP Algorithm

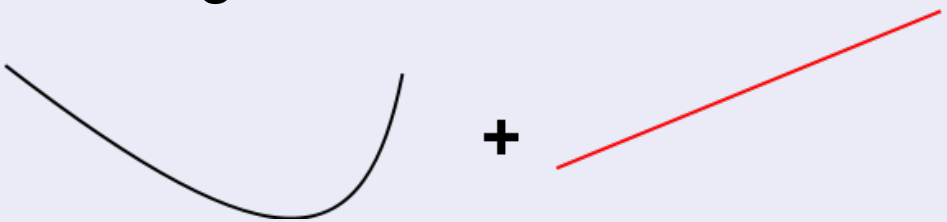
Concave-Convex Procedure (CCCP)

1. Decompose the objective into convex and concave part

2. Upper bound the concave part with a hyperplane



3. Minimize the resulting convex sum. Iterate until convergence



Non-Convex Objective - The CCCP Algorithm

- Our objective can be written as:

$$\min_{\vec{w}} \underbrace{\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y},\hat{h})]\right]}_{convex}$$

$$\underbrace{-\left[C\sum_{i=1}^{n} \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)\right]}_{concave}$$

Non-Convex Objective - The CCCP Algorithm

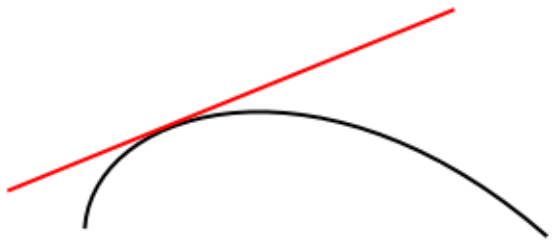
- Our objective can be written as:

$$\min_{\vec{w}} \underbrace{\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y},\hat{h})]\right]}_{convex}$$

$$\underbrace{-\left[C\sum_{i=1}^{n} \max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i,y_i,h)\right]}_{concave}$$

- Computing the upper bounding hyperplane is the same as completing the latent variables:

$$h_i^* = \text{argmax}_{h\in\mathcal{H}}\, \vec{w}\cdot\Phi(x_i,y_i,h)$$



Non-Convex Objective - The CCCP Algorithm

- Filling in the latent variables makes the objective convex

$$\min_{\vec{w}} \underbrace{\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y},\hat{h})]\right]}_{convex}$$

$$\underbrace{-\left[C\sum_{i=1}^{n} \vec{w}\cdot\Phi(x_i,y_i,h_i^*)\right]}_{linear}$$

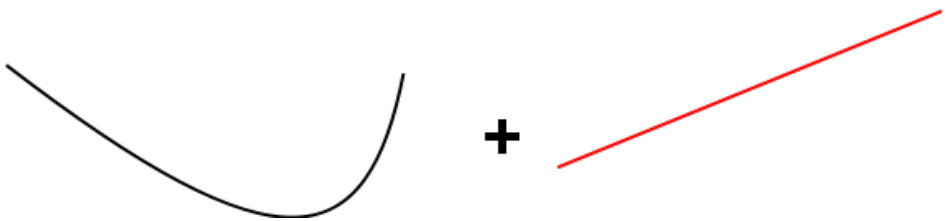
Non-Convex Objective - The CCCP Algorithm

- Filling in the latent variables makes the objective convex

$$\min_{\vec{w}} \underbrace{\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \max_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}} [\vec{w}\cdot\Phi(x_i,\hat{y},\hat{h}) + \Delta(y_i,\hat{y},\hat{h})]\right]}_{convex}$$

$$\underbrace{-\left[C\sum_{i=1}^{n} \vec{w}\cdot\Phi(x_i,y_i,h_i^*)\right]}_{linear}$$

- Cutting plane algorithm to solve convex optimization problem



Summary of Latent Structural SVM

Solve the optimization problem:

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Restriction on the loss function Δ

Summary of Latent Structural SVM

Solve the optimization problem:

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Restriction on the loss function Δ
- Three related inference problems:

Prediction:	$\text{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}}\ \vec{w}\cdot\Phi(x_i, y, h)$
Loss-augmented:	$\text{argmax}_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}}[\vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) + \Delta(y_i, \hat{y}, \hat{h})]$
Variable Completion:	$\text{argmax}_{h\in\mathcal{H}}\ \vec{w}\cdot\Phi(x_i, y_i, h)$

Summary of Latent Structural SVM

Solve the optimization problem:

$$\min_{\vec{w},\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$s.t.$ for $1 \leq i \leq n$, for all output structures $\hat{y} \in \mathcal{Y}$,

$$\max_{h\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, y_i, h) - \max_{\hat{h}\in\mathcal{H}} \vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) \geq \Delta(y_i, \hat{y}, \hat{h}) - \xi_i$$

- Restriction on the loss function Δ
- Three related inference problems:

Prediction:	$\text{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}}\ \vec{w}\cdot\Phi(x_i, y, h)$
Loss-augmented:	$\text{argmax}_{(\hat{y},\hat{h})\in\mathcal{Y}\times\mathcal{H}}[\vec{w}\cdot\Phi(x_i, \hat{y}, \hat{h}) + \Delta(y_i, \hat{y}, \hat{h})]$
Variable Completion:	$\text{argmax}_{h\in\mathcal{H}}\ \vec{w}\cdot\Phi(x_i, y_i, h)$

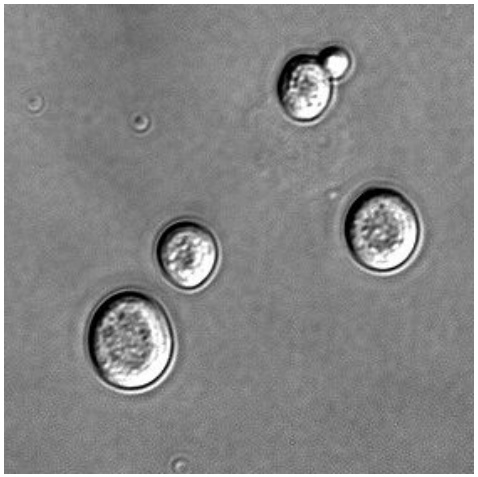
- No computation of partition functions required

Applications

- Discriminative motif finding
- Noun phrase coreference resolution

Discriminative Motif Finding

- Input x: DNA sequences containing ARS from S. cerevisiae and S. kluyveri



S. cerevisiae

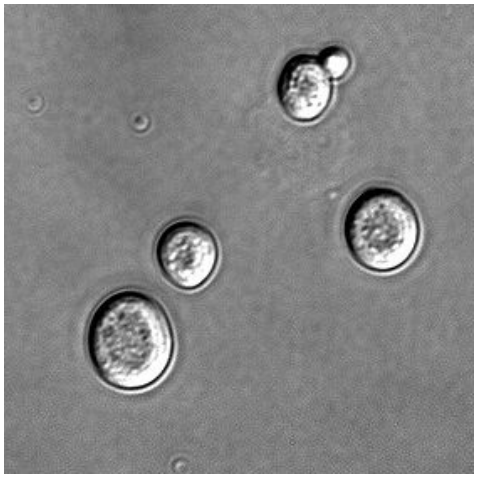
```
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
```

S. kluyveri

```
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
ACGTACGT.........ACGT
```

Discriminative Motif Finding

- Input x: DNA sequences containing ARS from S. cerevisiae and S. kluyveri



- Label y: Whether the sequence replicates in S. cerevisiae

S. cerevisiae

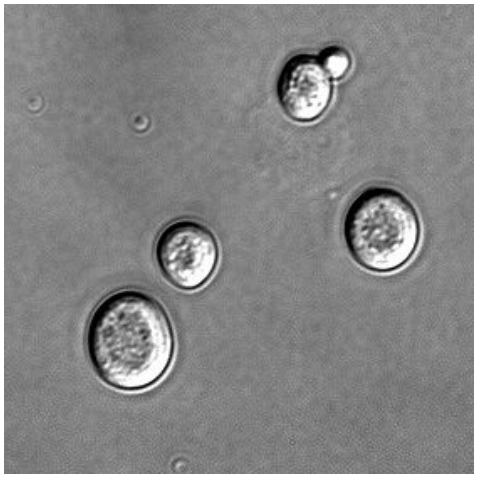
+ ACGTACGT.....ACGT
+ ACGTACGT.....ACGT
+ ACGTACGT.....ACGT
+ ACGTACGT.....ACGT
+ ACGTACGT.....ACGT

S. kluyveri

+ ACGTACGT.....ACGT
+ ACGTACGT.....ACGT
- ACGTACGT.....ACGT
- ACGTACGT.....ACGT
+ ACGTACGT.....ACGT

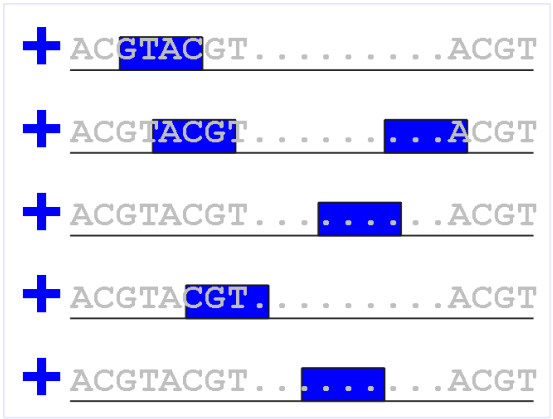
Discriminative Motif Finding

- Input x: DNA sequences containing ARS from S. cerevisiae and S. kluyveri

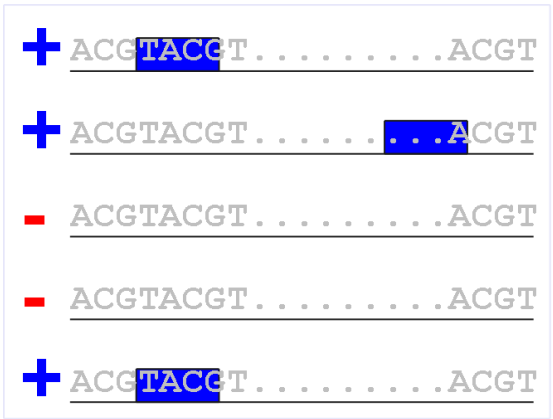


- Label y: Whether the sequence replicates in S. cerevisiae
- Latent variable h: position of the motif

S. cerevisiae

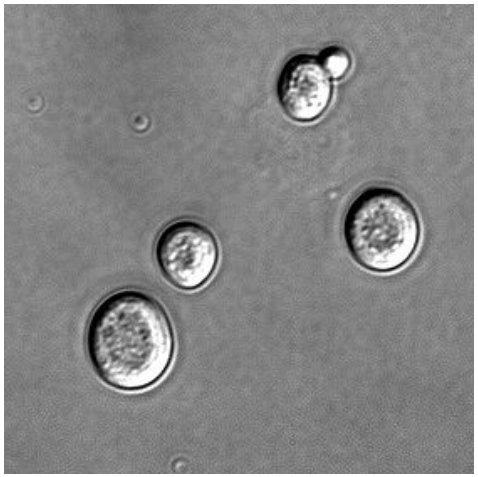


S. kluyveri



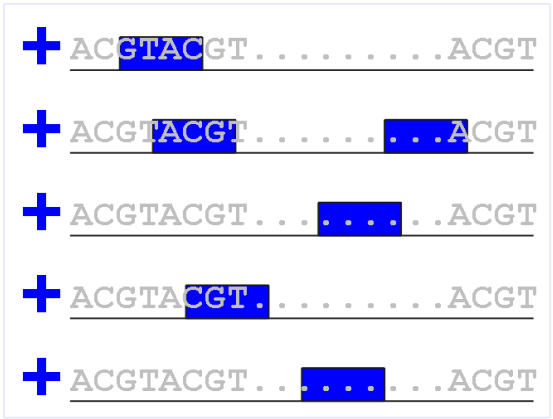
Discriminative Motif Finding

- Input x: DNA sequences containing ARS from S. cerevisiae and S. kluyveri

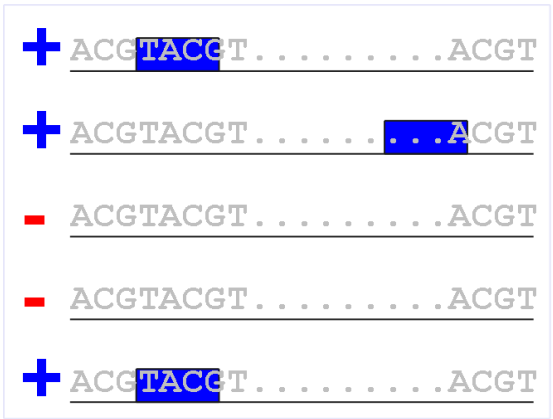


- Label y: Whether the sequence replicates in S. cerevisiae
- Latent variable h: position of the motif
- Task: Find out the predictive motif

S. cerevisiae



S. kluyveri



Discriminative Motif Finding - Formulation

- Feature vector Φ: Position-specific weight matrix plus parameters for Markov background model

$$\Phi(x, y, h) = \underbrace{\sum_{i=1}^{h} \phi_{BG}(x_i)}_{\text{background}} + \underbrace{\sum_{j=1}^{l} \phi_{PSM}^{(j)}(x_{h+j})}_{\text{motif}} + \underbrace{\sum_{i=h+l+1}^{n} \phi_{BG}(x_i)}_{\text{background}}$$

d Position weight matrix (PWM)

A	−1.93	0.79	0.79	−1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	−1.93	−1.93	0.79
C	0.45	−1.93	0.79	1.68	−1.93	−1.93	−1.93	0.45	−1.93	−1.93	−1.93	−1.93	0.00	0.79
G	0.00	0.45	−1.93	−1.93	−1.93	−1.93	−1.93	−1.93	0.66	−1.93	1.30	1.68	1.07	−1.93
T	0.15	0.66	−1.93	−1.93	1.07	0.66	0.79	0.00	0.00	0.79	−1.93	−1.93	−0.66	−1.93

[from Wasserman 2004]

Discriminative Motif Finding - Formulation

- Feature vector Φ: Position-specific weight matrix plus parameters for Markov background model

$$\Phi(x, y, h) = \underbrace{\sum_{i=1}^{h} \phi_{BG}(x_i)}_{\text{background}} + \underbrace{\sum_{j=1}^{l} \phi_{PSM}^{(j)}(x_{h+j})}_{\text{motif}} + \underbrace{\sum_{i=h+l+1}^{n} \phi_{BG}(x_i)}_{\text{background}}$$

d Position weight matrix (PWM)

A	−1.93	0.79	0.79	−1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	−1.93	−1.93	0.79
C	0.45	−1.93	0.79	1.68	−1.93	−1.93	−1.93	0.45	−1.93	−1.93	−1.93	−1.93	0.00	0.79
G	0.00	0.45	−1.93	−1.93	−1.93	−1.93	−1.93	−1.93	0.66	−1.93	1.30	1.68	1.07	−1.93
T	0.15	0.66	−1.93	−1.93	1.07	0.66	0.79	0.00	0.00	0.79	−1.93	−1.93	−0.66	−1.93

[from Wasserman 2004]

- Loss function Δ: Zero-one loss

Discriminative Motif Finding - Formulation

- Feature vector Φ: Position-specific weight matrix plus parameters for Markov background model

$$\Phi(x, y, h) = \underbrace{\sum_{i=1}^{h} \phi_{BG}(x_i)}_{\text{background}} + \underbrace{\sum_{j=1}^{l} \phi_{PSM}^{(j)}(x_{h+j})}_{\text{motif}} + \underbrace{\sum_{i=h+l+1}^{n} \phi_{BG}(x_i)}_{\text{background}}$$

d Position weight matrix (PWM)

A	−1.93	0.79	0.79	−1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	−1.93	−1.93	0.79
C	0.45	−1.93	0.79	1.68	−1.93	−1.93	−1.93	0.45	−1.93	−1.93	−1.93	−1.93	0.00	0.79
G	0.00	0.45	−1.93	−1.93	−1.93	−1.93	−1.93	−1.93	0.66	−1.93	1.30	1.68	1.07	−1.93
T	0.15	0.66	−1.93	−1.93	1.07	0.66	0.79	0.00	0.00	0.79	−1.93	−1.93	−0.66	−1.93

[from Wasserman 2004]

- Loss function Δ: Zero-one loss
- Inference: enumeration, as y is binary and h is linear in sequence length

Discriminative Motif Finding - Results

- Data - 197 yeast DNA sequences from S. cerevisiae and S. kluyveri.

Discriminative Motif Finding - Results

- Data - 197 yeast DNA sequences from S. cerevisiae and S. kluyveri.
- $\sim$6000 intergenic sequences for background estimation

Discriminative Motif Finding - Results

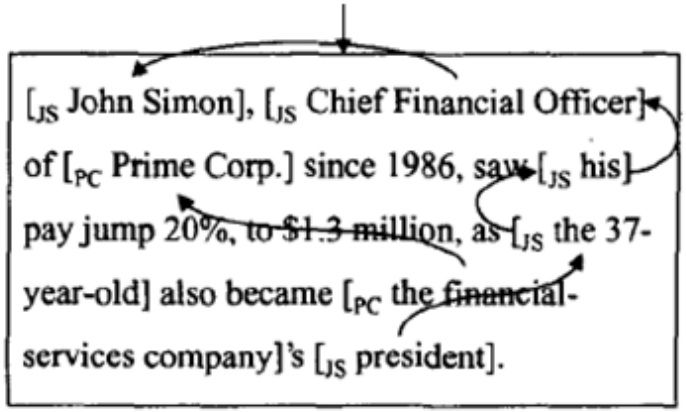
- Data - 197 yeast DNA sequences from S. cerevisiae and S. kluyveri.
- $\sim$6000 intergenic sequences for background estimation
- 10-fold CV, 10 random restarts for each parameter setting

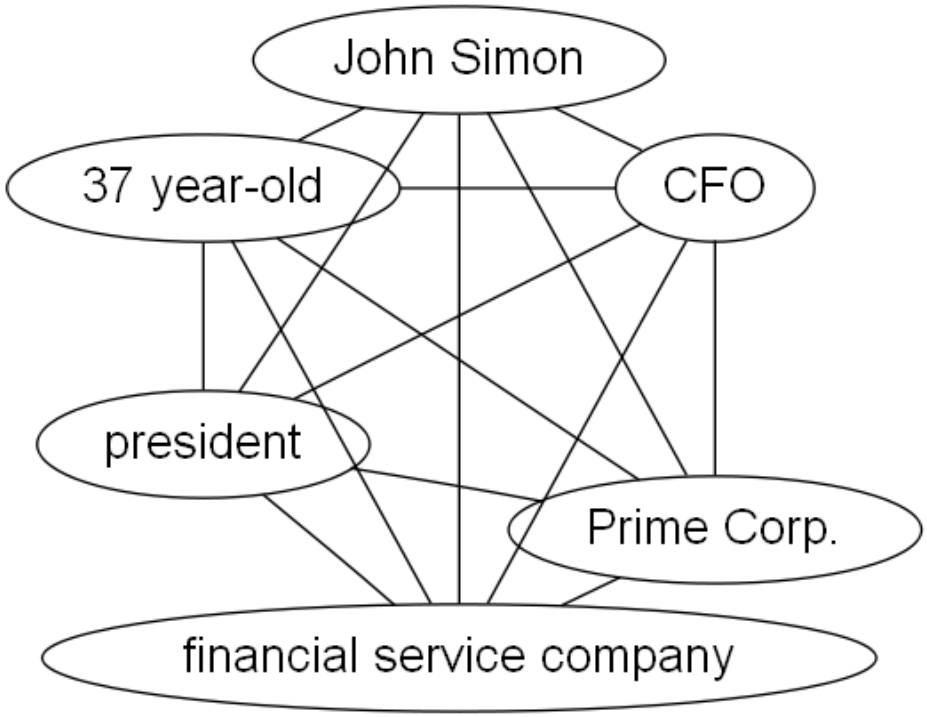
Algorithm	Error Rate
Gibbs Sampler (w=11)	37.9%
Gibbs Sampler (w=17)	35.06%
Latent Structural SVM (w=11)	11.09%
Latent Structural SVM (w=17)	12.00%

Noun Phrase Coreference

- Input x: Noun phrases with edge features

[from Cardie & Wagstaff '99]

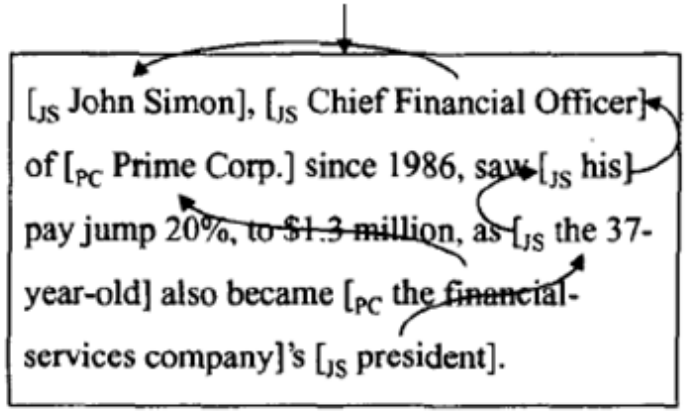


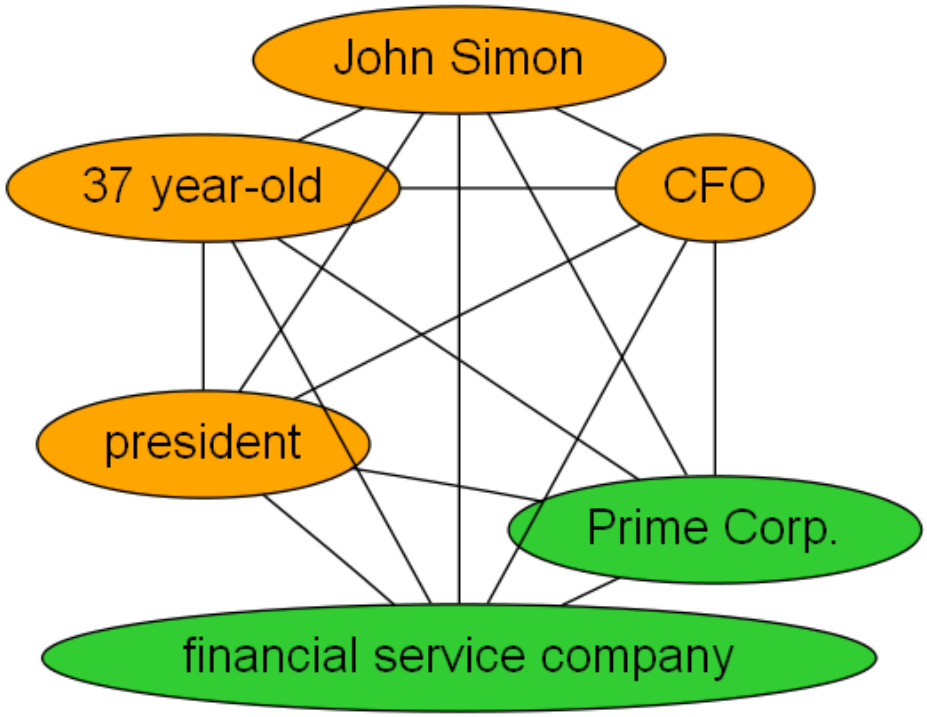


Noun Phrase Coreference

- Input x: Noun phrases with edge features
- Label y: Clusters of noun phrases

[from Cardie & Wagstaff '99]

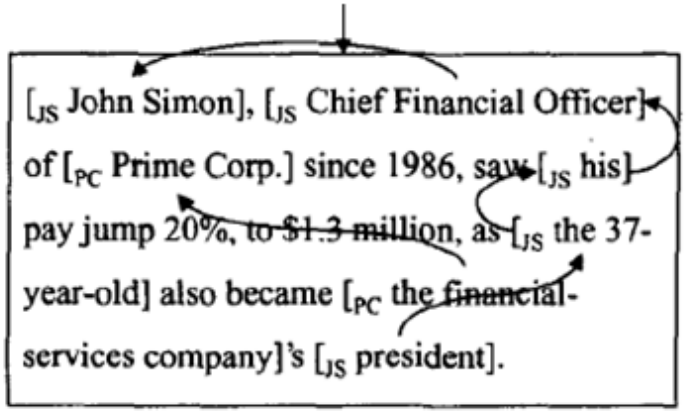


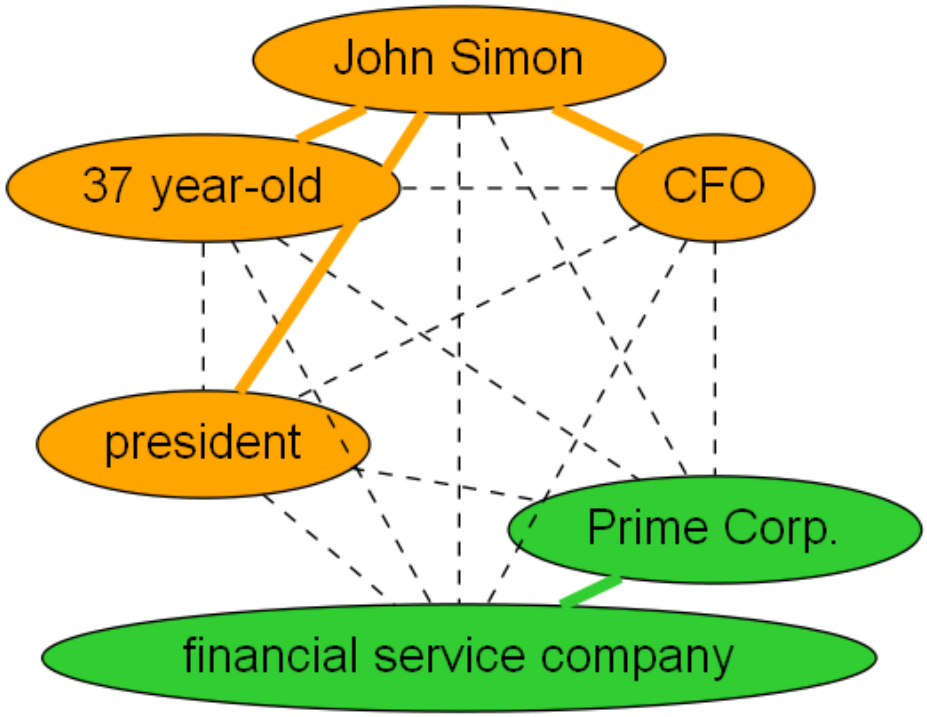


Noun Phrase Coreference

- Input x: Noun phrases with edge features
- Label y: Clusters of noun phrases
- Latent variable h: 'Strong' links as trees

[from Cardie & Wagstaff '99]

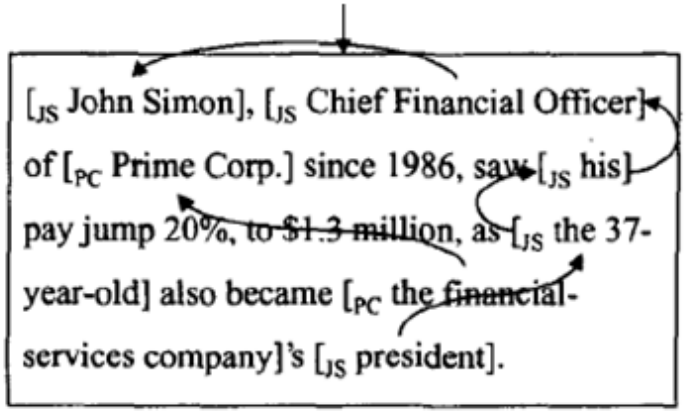


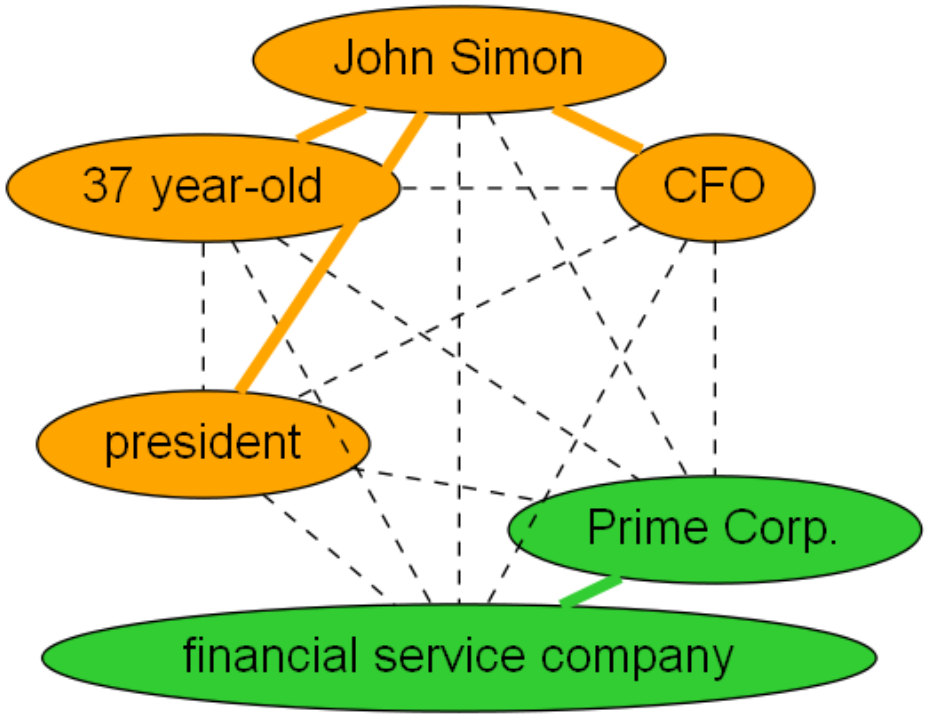


Noun Phrase Coreference

- Input x: Noun phrases with edge features
- Label y: Clusters of noun phrases
- Latent variable h: 'Strong' links as trees
- Task: Cluster the noun phrases using single-link agglomerative clustering

[from Cardie & Wagstaff '99]

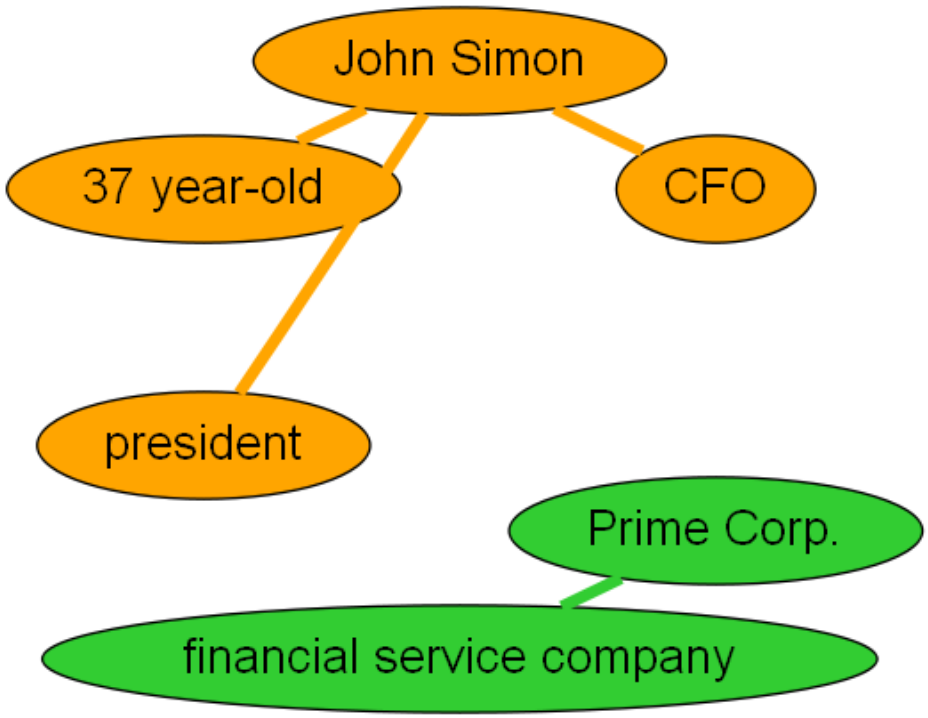




Noun Phrase Coreference - Formulation

- Feature vector Φ: sum of tree edge features:

$$\Phi(x, y, h) = \sum_{(i,j) \in h} x_{ij}$$



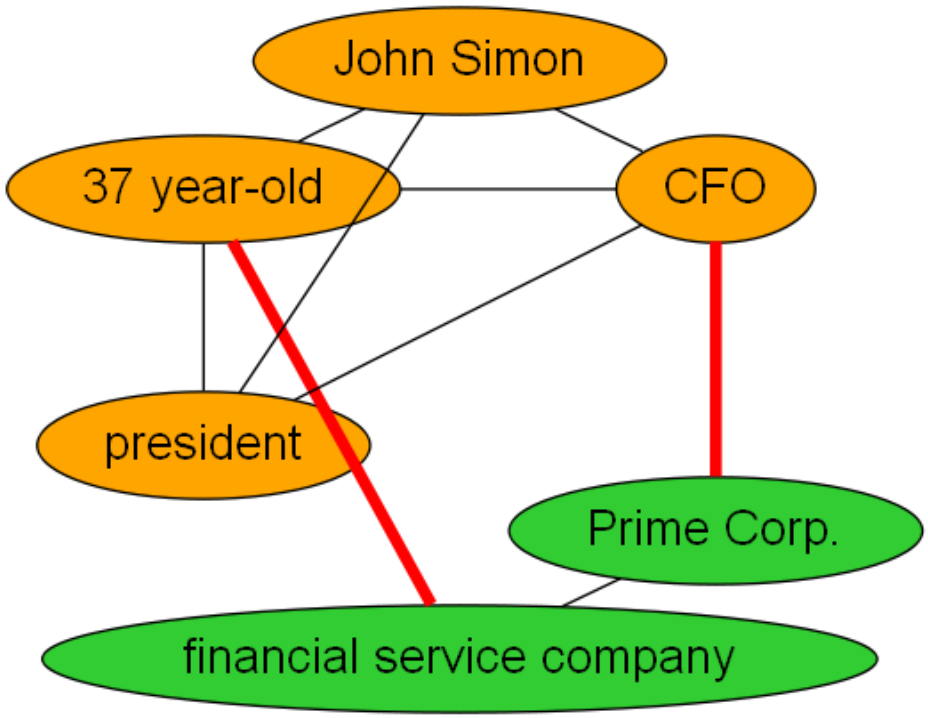
Noun Phrase Coreference - Formulation

- Feature vector Φ: sum of tree edge features:

$$\Phi(x, y, h) = \sum_{(i,j) \in h} x_{ij}$$

- Loss function Δ:

$$\Delta(y, \hat{y}, \hat{h}) = \underbrace{n(y)}_{\#nodes} - \underbrace{k(y)}_{\#components} + \sum_{(i,j) \in \hat{h}} \underbrace{\ell(y, (i,j))}_{+1/-1}$$



Noun Phrase Coreference - Formulation

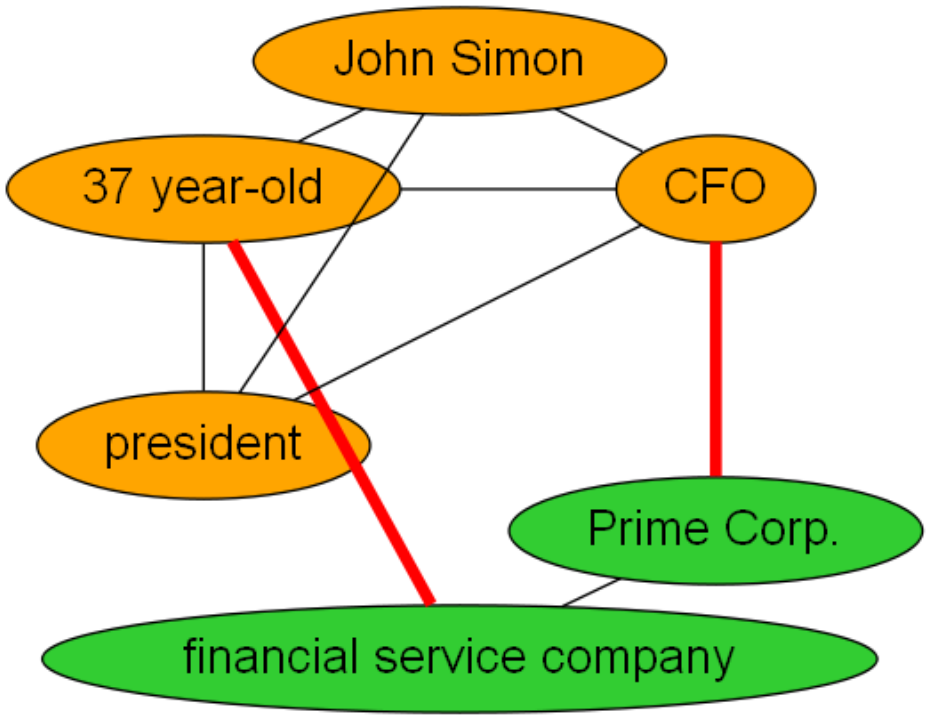
- Feature vector Φ: sum of tree edge features:

$$\Phi(x, y, h) = \sum_{(i,j) \in h} x_{ij}$$

- Loss function Δ:

$$\Delta(y, \hat{y}, \hat{h}) = \underbrace{n(y)}_{\#nodes} - \underbrace{k(y)}_{\#components} + \sum_{(i,j) \in \hat{h}} \underbrace{\ell(y, (i,j))}_{+1/-1}$$

- Inference: Any Maximum Spanning Tree algorithm



Noun Phrase Coreference - Results

- Test on MUC 6 data, using the same features as in [Ng & Cardie '02]

Noun Phrase Coreference - Results

- Test on MUC 6 data, using the same features as in [Ng & Cardie '02]
- Initialize spanning trees by chronological order

Noun Phrase Coreference - Results

- Test on MUC 6 data, using the same features as in [Ng & Cardie '02]
- Initialize spanning trees by chronological order
- 10-fold CV results:

Algorithm	MITRE loss	Pair loss
$SVM^{cluster}$ [Finley & Joachims '05]	41.3	2.89
Latent SSVM	44.1	2.66
Latent SSVM (asymmetric loss)	35.6	4.11

Conclusions

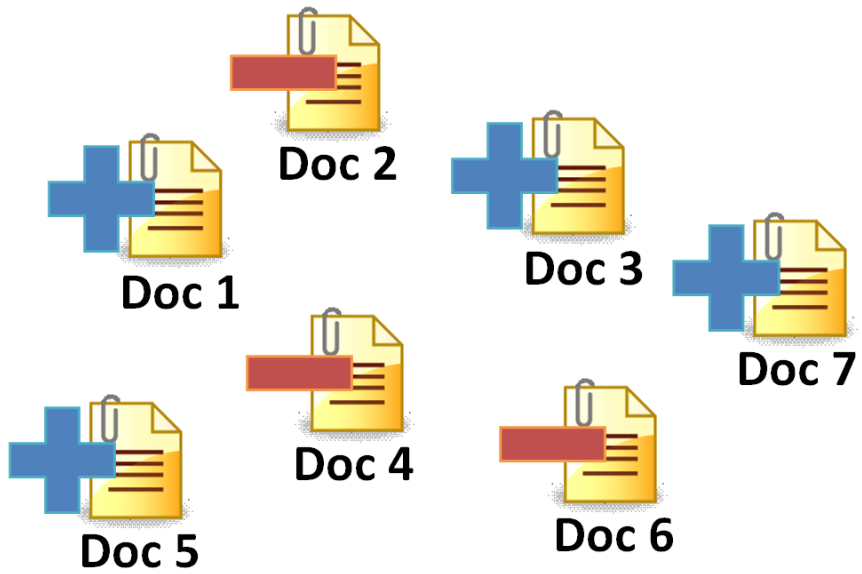
- A formulation of structural SVM that can handle latent variables
- Efficient solution algorithm for the formulation
- Demonstrated the use of the algorithm on three applications, with good prediction performance
- Modular design allows us to focus only on the design of feature vector, loss function, and inference procedures
- Link to software package:
  `http://www.cs.cornell.edu/~cnyu/latentssvm/`

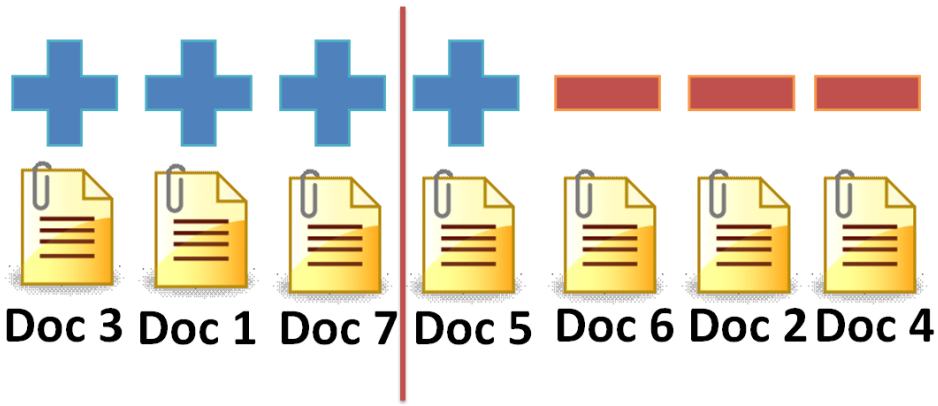
Thank you!

Optimizing Precision@k

- Input x: A query with an associated collection of documents
- Label y: Relevance judgments of each document
- Latent variable h: Top k relevant documents

Query q: ICML 2009





Optimizing Precision@k - Formulation

- Feature vector Φ: sum of features from top k documents

$$\Phi(x, y, h) = \sum_{j=1}^{k} x_{h_j}$$

- Loss function Δ: One minus precison@k

$$\Delta(y, \hat{y}, \hat{h}) = 1 - \frac{1}{k} \sum_{j=1}^{k} [y_{\hat{h}_j} == 1]$$

- Depends only on top k document selected by h
- Inference: Sorting

Optimizing Precision@k - Results

- OHSUMED dataset from LETOR 3.0 benchmark
- Initialize *h* with weight vector trained on classification accuracy
- 5-fold CV results: