Structured learning

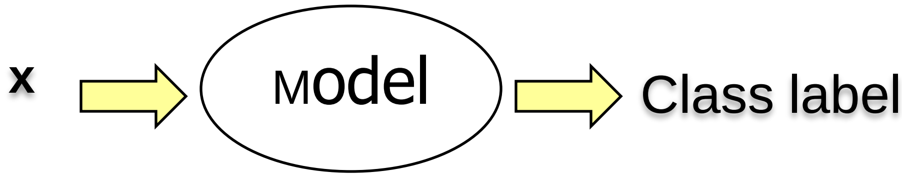
Sunita Sarawagi

IIT Bombay

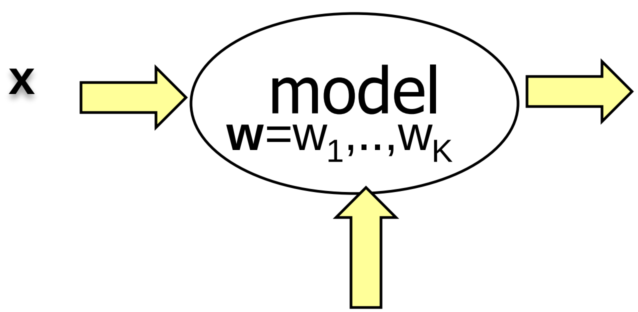
http://www.cse.iitb.ac.in/~sunita

Structured models

- Standard classification



- Structured prediction



Structured y

1. Vector: y_1 ,y_2,..,y_n
2. Segmentation
3. Tree
4. Alignment
5. ..

Feature function vector

$\mathbf{f(x,y)} = f_1(\mathbf{x,y}), f_{\mathbf{2}}(\mathbf{x,y}),\ldots,f_K(\mathbf{x,y}),$

Structured model

- Score of a prediction **y** for input **x**:
  - $s(\mathbf{x},\mathbf{y}) = \mathbf{w}.\ \mathbf{f}(\mathbf{x},\mathbf{y})$
- Prediction problem: find highest scoring output
  - $\mathbf{y}_* = \operatorname{argmax}_{\mathbf{y}}\ s(\mathbf{x},\mathbf{y})$
  - Space of possible **y** exponentially large
  - Exploit decomposability of feature functions
    - $\mathbf{f}(\mathbf{x},\mathbf{y}) = \sum_c \mathbf{f}\,(\mathbf{x},\mathbf{y}_c,c)$
- Training problem: find **w** given many correct input-output pairs $(\mathbf{x}_1\ \ \mathbf{y}_1), (\mathbf{x}_2\ \ \mathbf{y}_2), \ldots, (\mathbf{x}_N\ \ \mathbf{y}_N)$

Outline

- Applications
- Inference algorithms
- Training objectives and algorithms

Information Extraction (IE)

- Find structure in unstructured text

According to Robert Callahan, president of Eastern's flight attendants union, the past practice of Eastern's parent, Houston-based Texas Air Corp., has involved ultimatums to unions to accept the carrier's terms

Author Year Title Journal Volume Page

P.P.Wangikar, T.P. Graycar, D.A. Estell, D.S. Clark, J.S. Dordick (1993) Protein and Solvent Engineering of Subtilising BPN' in Nearly Anhydrous Organic Media J.Amer. Chem. Soc. 115, 12231-12237.

Others

- Disease outbreaks from news articles
- Addresses/Qualifications from resumes for HR DBs
- Room attributes from hotel websites
- Proteins and interactions from bio-medical abstracts

Clues that drive extraction

- Orthographic patterns: names have two capitalized words.
- Keywords: “In” is within 1—3 tokens before location
- Order of entities: Titles appear before Journal names
- Position: Product titles follow a N(4in,1) distance from top
- Dictionary match: Authors have high similarity with person_name column of DB
- Collective: All occurrences of a word prefer the same label

Learning models for IE

- Rule-based models (1980s)
  - Too brittle, not for noisy environment.
- Classifiers for boundaries (1980s)
  - Could give inconsistent labels
- Hidden Markov Models (1990s)
  - Generative model, restrictive features
- Maxent Taggers (1990s) & MeMMs (late 1990)
  - Label bias problem.
- Conditional Random Fields (2000s)
- Segmentation models. (2004)

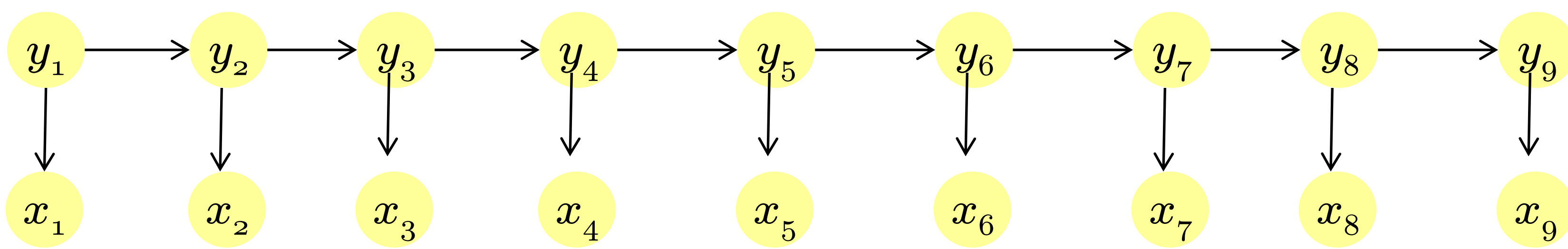
Sequence labeling

$\boxed{\text{My review of Fermat's last theorem by S. Singh}}$

Sequence labeling

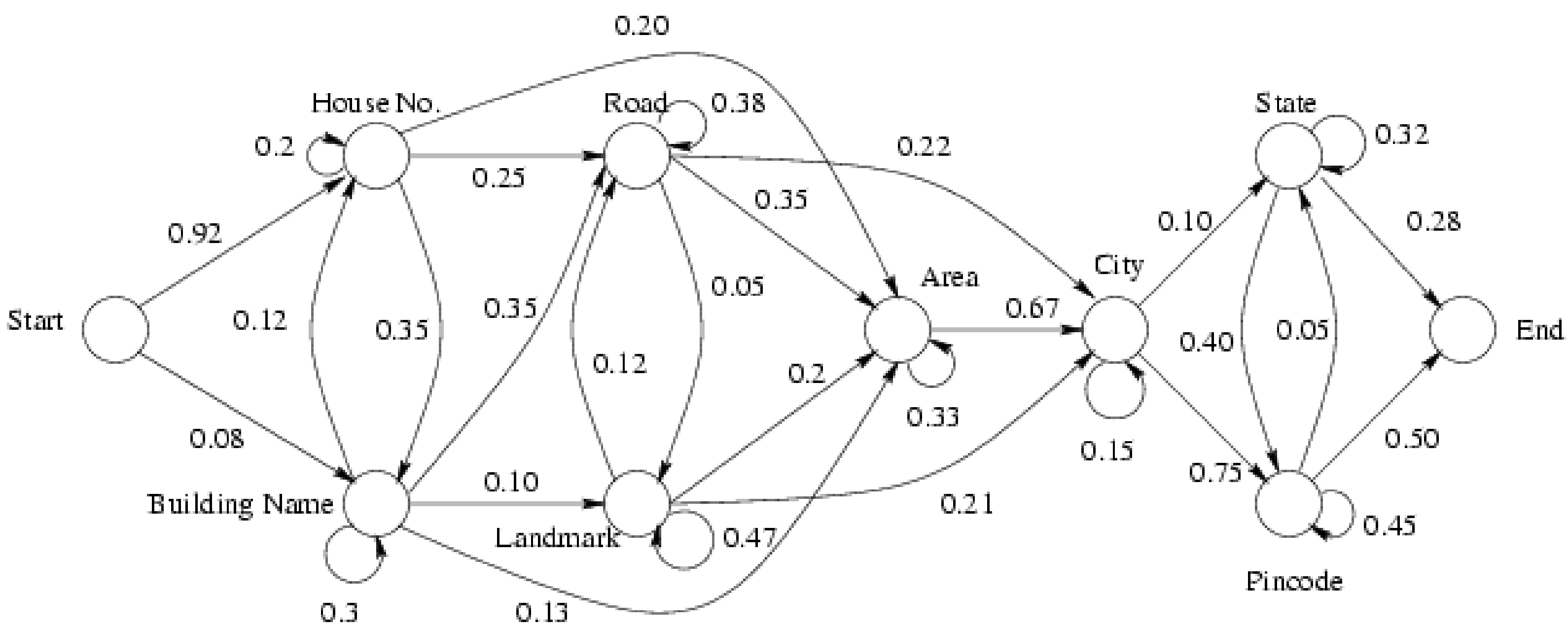
My review of Fermat's last theorem by S. Singh

t	1	2	3	4	5	6	7	8	9
x	My	review	of	Fermat's	last	theorem	by	S.	Singh
y	**Other**	**Other**	**Other**	**Title**	**Title**	**Title**	**other**	**Author**	**Author**



HMM for IE

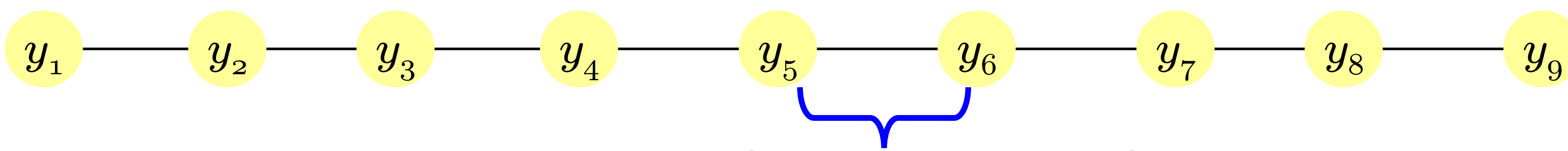
- The two types of parameters
  - $\Pr(x_i \mid y_i)$
    - Multinomial distribution of words in each state
  - $\Pr(y_i \mid y_{i-1})$



Structured learning for IE

My review of Fermat's last theorem by S. Singh

t	1	2	3	4	5	6	7	8	9
x	My	review	of	Fermat's	last	theorem	by	S.	Singh
y	**Other**	**Other**	**Other**	**Title**	**Title**	**Title**	**other**	**Author**	**Author**



$$\mathbf{f}(y_i, y_{i-1}, i, \mathbf{x})$$

Features decompose over adjacent labels.

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \mathbf{f}(y_i, y_{i-1}, i, \mathbf{x})$$

MAP can be found in $O(nm^2)$ time

Features

- Feature vector for each position

$$\mathbf{f}(y_i, \mathbf{x}, i, y_{i-1})$$

User provided

i-th label

Word i & neighbors

previous label

- Examples

$$f_2(y_i, \mathbf{x}, i, y_{i-1}) = 1 \text{ if } y_i \text{ is Person \& } x_i \text{ is Douglas}$$

$$f_3(y_i, \mathbf{x}, i, y_{i-1}) = 1 \text{ if } y_i \text{ is Person \& } y_{i-1} \text{ is Other}$$

Features in typical extraction tasks

- Words
- Orthographic word properties
  - Capitalized? Digit? Ends-with-dot?
- Part of speech
  - Noun?
- Match in a dictionary
  - Appears in a dictionary of people names?
  - Appears in a list of stop-words?

- Fire these for each label and
  - The token,
  - W tokens to the left or right, or
  - Concatenation of tokens.

Publications

- Cora dataset
  - Paper headers: Extract title,author affiliation, address,email,abstract
    - 94% F1 with CRFs
    - 76% F1 with HMMs
  - Paper citations: Extract title,author,date, editor,booktitle,pages,institution
    - 91% F1 with CRFs
    - 78% F1 with HMMs

Peng & McCallum 2004

IE as Segmentation

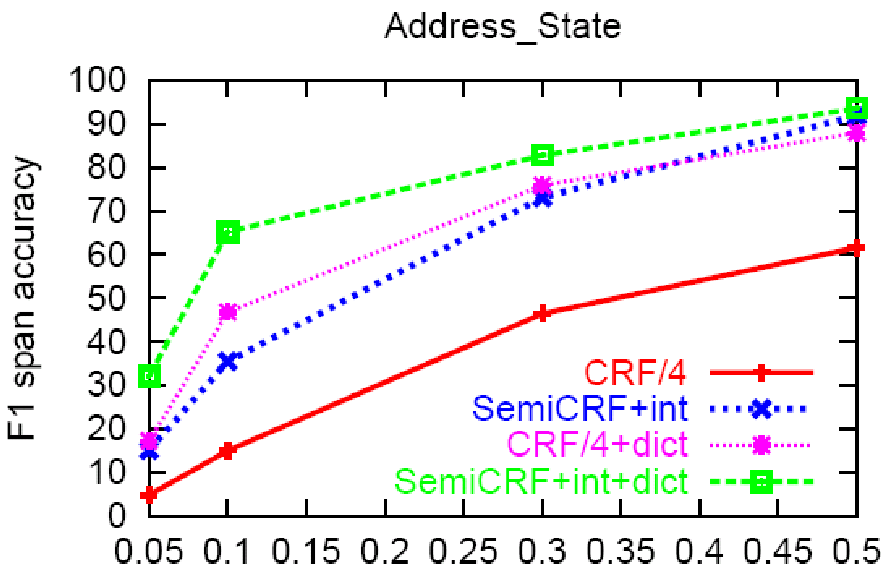
x	My	review	of	Fermat's	last	theorem	by	S.	Singh
y	**Other**	**Other**	**Other**	**Title**			**other**	**Author**	

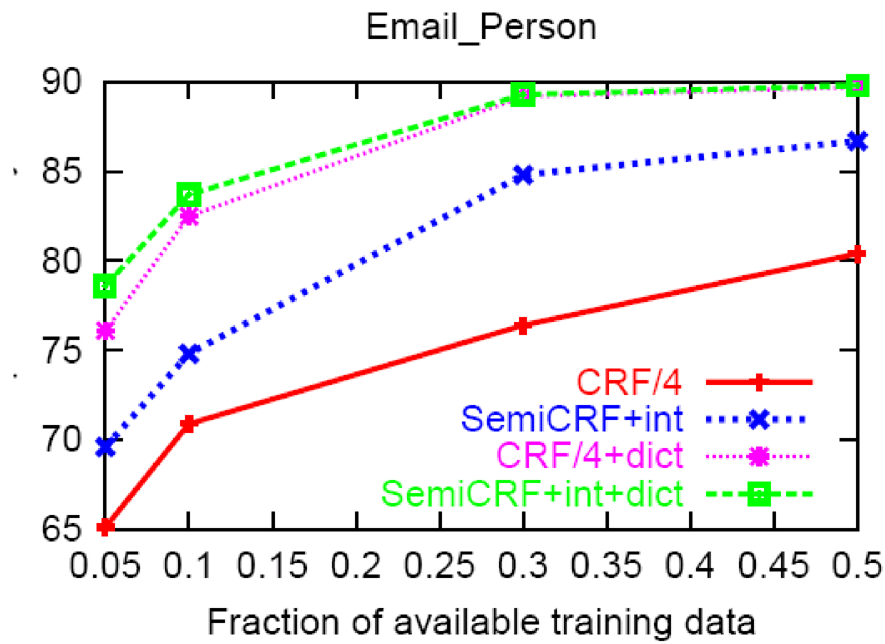
- Output **y** is a sequence of segments $s_1,\ldots,s_p$
- Feature f(**x**,**y**) decomposes over segment and label of previous segment

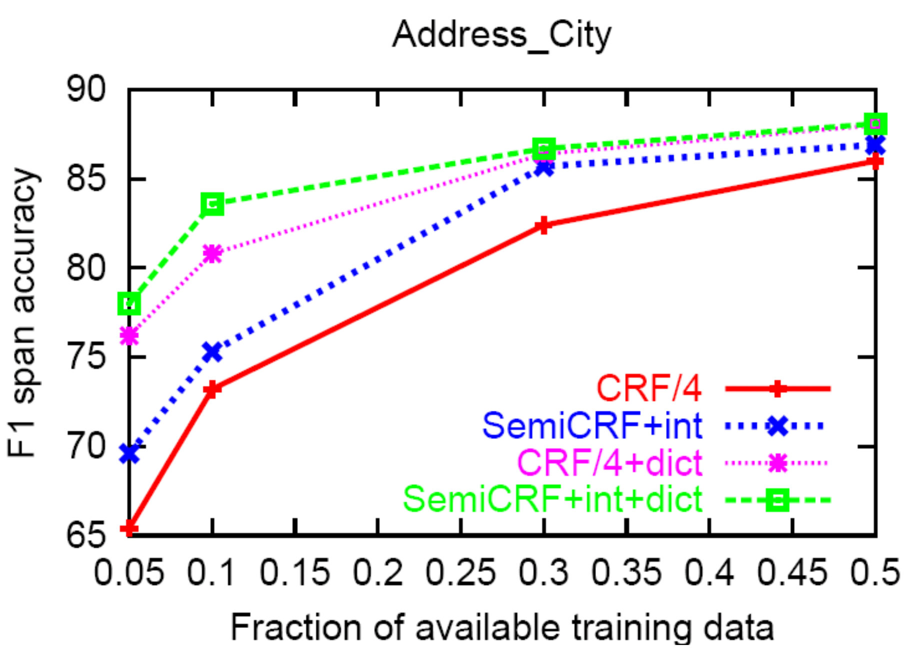
$$\mathbf{f}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{p} \mathbf{f}(\mathbf{x}, s_j, y_{j-1})$$

- MAP: easy extension of Viterbi $O(m^2 n^2)$
  - m = number of labels, n = length of a sequence

Some Results



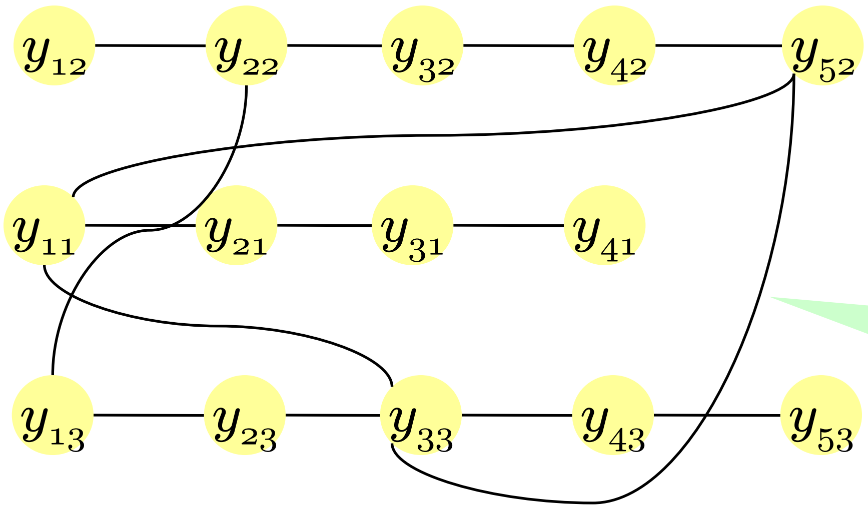




- **CRF/4 – baseline CRF method**
- **SemiCRF+int – semiCRF with internal dictionary features**
- **CRF/4+dict – baseline + distance of tokens to an external dictionary**
- **SemiCRF+int+dict – semiCRF with all features, including external dictionary-based features**

Collective labeling

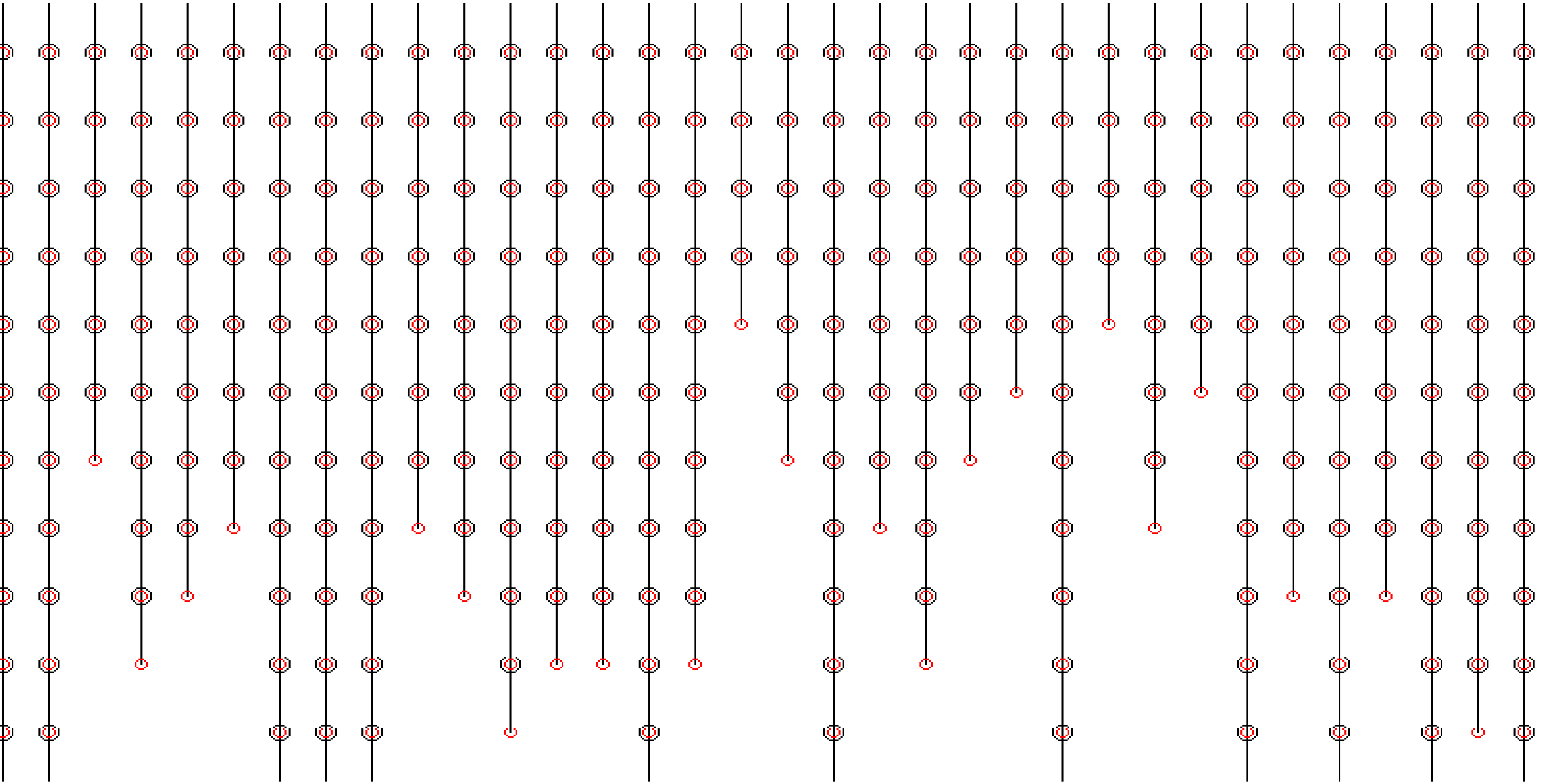
- Y does have character.
- Mr. X lives in Y.
- X buys Y Times daily.



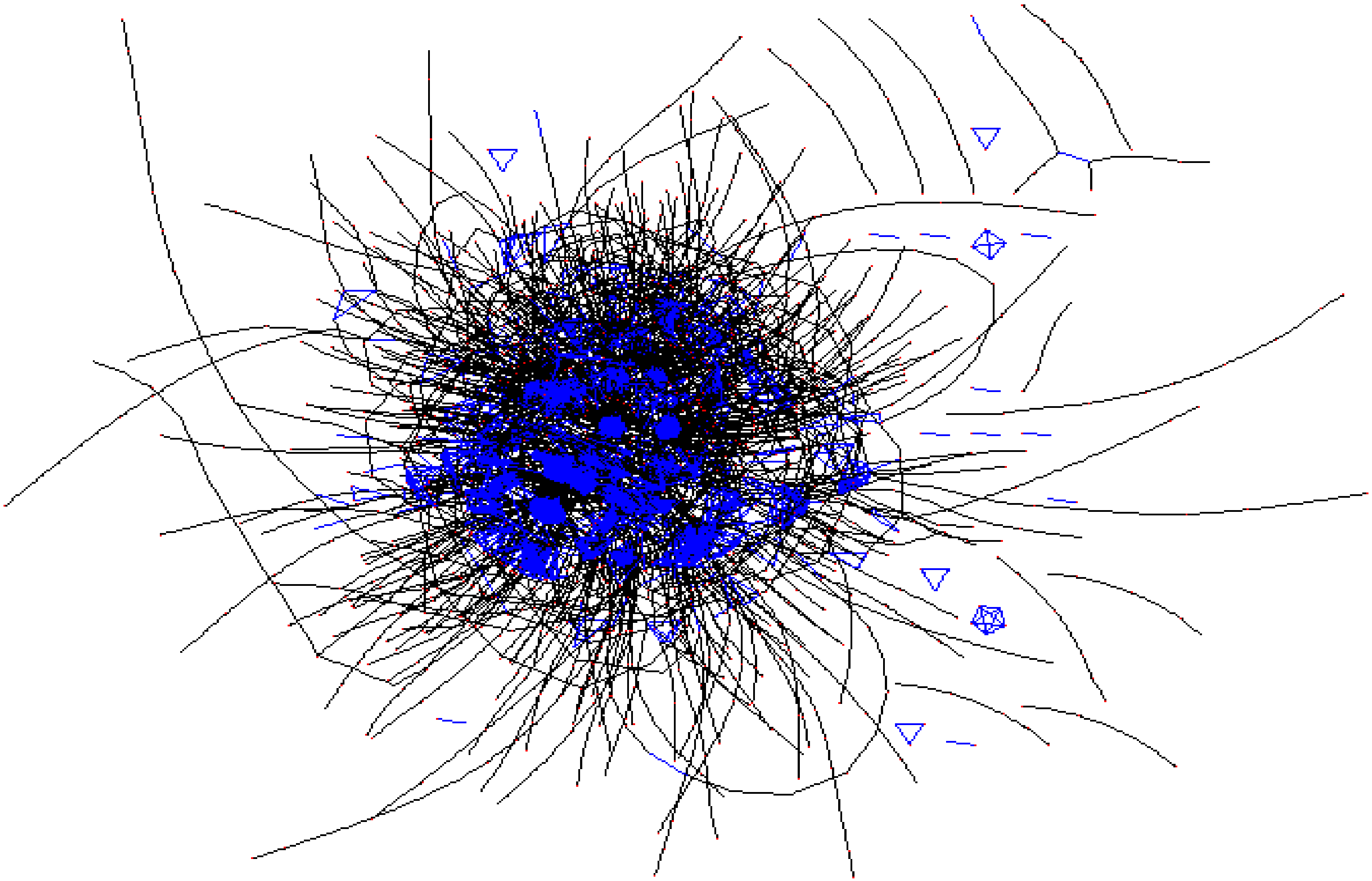
Associative potentials
$\phi_e(i,i) > \phi_e(i,j)$

$$\mathbf{f}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{D} \sum_{j=1}^{|D_i|} \mathbf{f}(\mathbf{x}, y_{ij}, y_{ij-1}, i) + \sum_{x_{ij}=x_{i'j'}} f_e(y_{ij}, y_{i'j'})$$

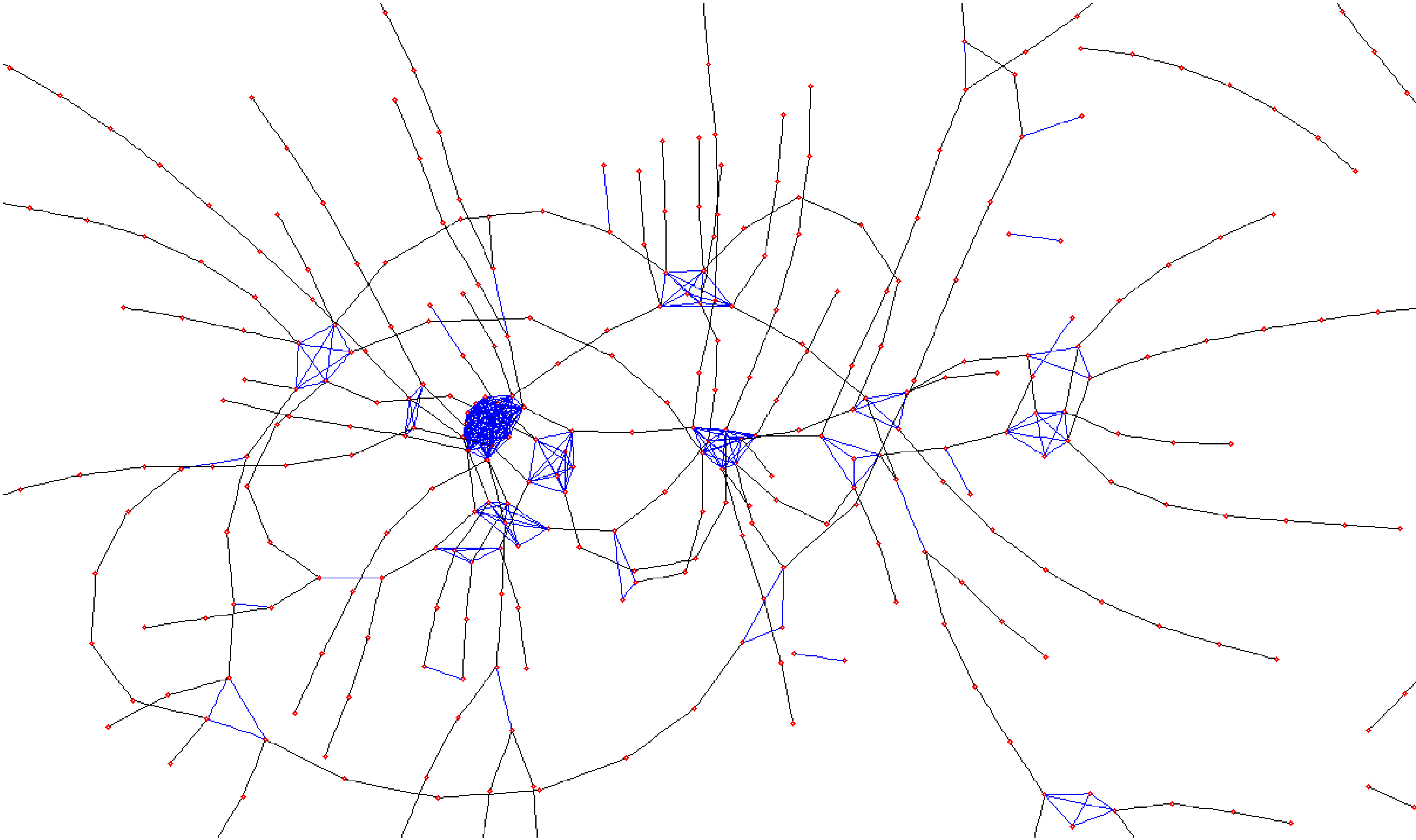
Starting graphs (..of an extraction task from addresses)



Graph after collective edges

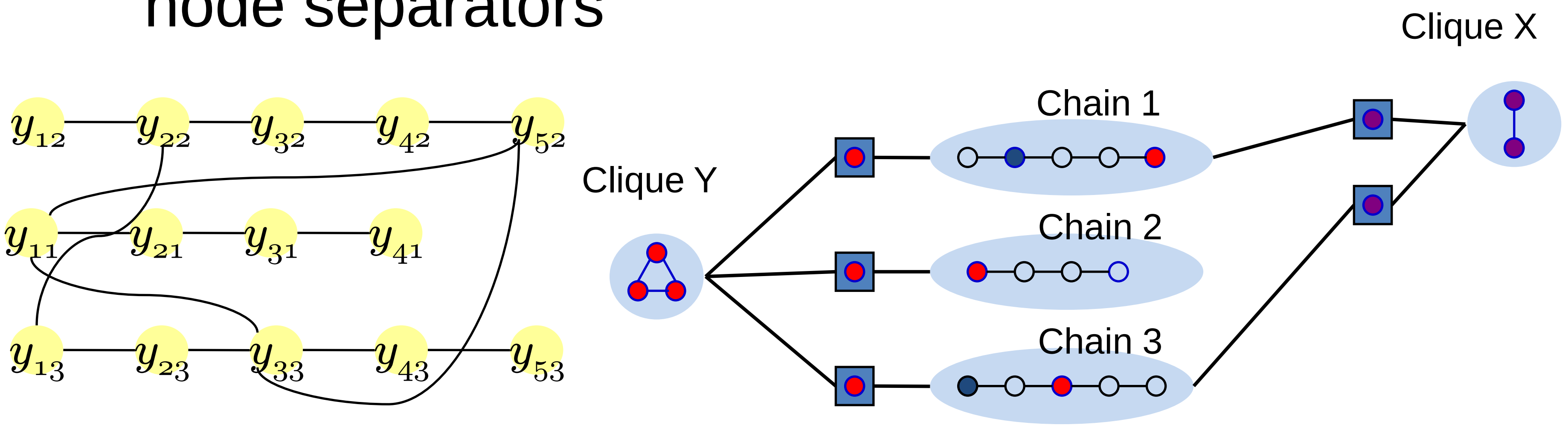


A closer look at the graph…



Our approach

BP on clusters of cliques and chains with single node separators



- Basic MP step: Compute max-marginals for a separator node ➔ MAP for each label of the node.
- MAP algorithms for chains ➔ easy and efficient.
- MAP algorithms for cliques➔ Design new combinatorial algorithms

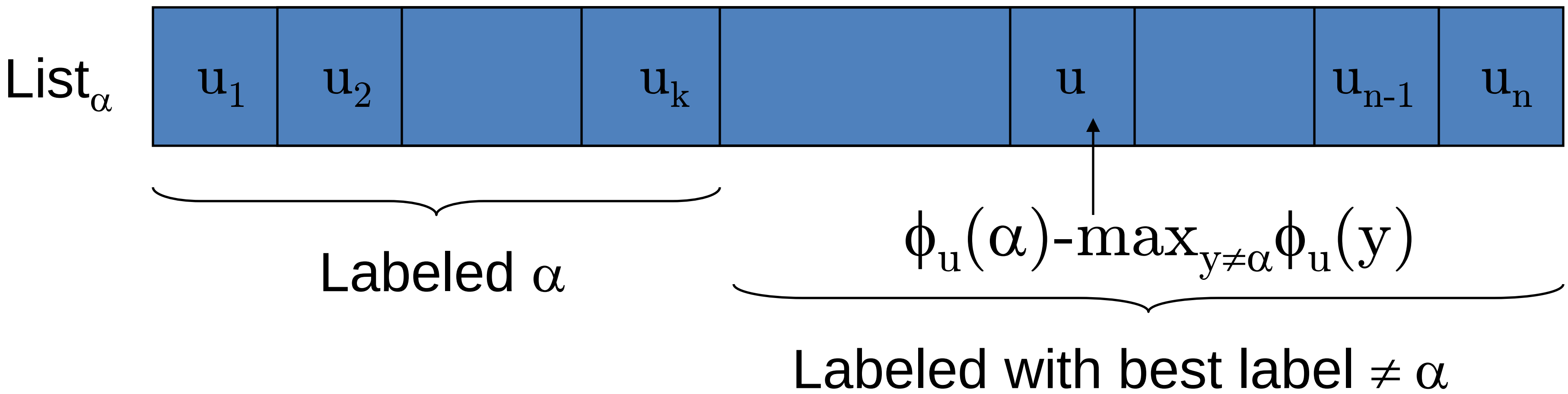
Clique inference

- Given a clique c with n nodes, m labels
  - $\phi_{\mathbf{u}}(\mathbf{y_u})$ Node Potentials for each node u $\in$ c
  - $\mathbf{cp}(\mathbf{y})$ Clique Potential over all nodes in c
- Find MAP labeling **y*** as
  - $\mathbf{y}^* = \text{argmax}_{\mathbf{y}} \left(\sum_{\mathbf{u}} \phi_{\mathbf{u}}(\mathbf{y_u}) + \mathbf{cp}(\mathbf{y})\right)$
- Two properties of clique potentials
  - Associative
  - Symmetric: depends only on label counts
    - $CP(y_1,\ldots,y_n) = f(n(y)) = f(n_1,n_2,\ldots,n_m)$

Cardinality-based Clique Potentials

MAX	$\max_y f(n_y)$	Optimal
MAJ	$\sum_y w_{\alpha y} n_y$ $(\alpha = \operatorname{argmax}_y n_y)$	Optimal based on Lagrange relaxation
SUM	$\sum_y n_y^2$ (POTTS) $\sum_y n_y \log n_y$ (Entropy)	13/15 Approx ½ Approx O(nlog n) time.

The α–pass Algorithm

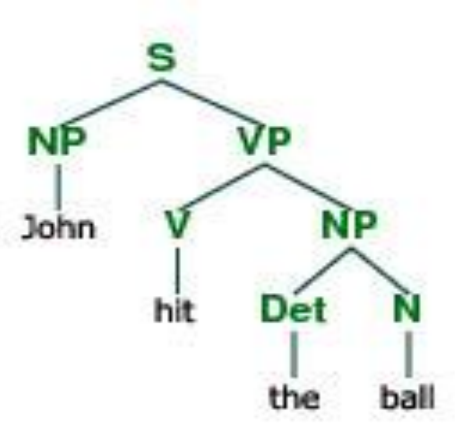


1. For every α, sort nodes by $\phi_u(\alpha)\text{-}\max_{y\neq\alpha}\phi_u(y)$
    1. For all $1 \leq k \leq n$
        1. Label first k nodes with α
        2. Label the rest with their best non-α labels.
2. Pick the best solution across all (α, k) combinations.

$O(mn \log n)$

Parse tree of a sentence

- Input **x**: “John hit the ball”
- Output **y**: parse tree



- Features decompose over nodes of the tree
- MAP: Inside/outside algorithm $O(n^3)$

- Better than Probabilistic CFGs (Taskar EMNLP 2004)

Sentence alignment

- Input **x**: sentence pair
- Output **y** : alignment

  - $y_{i,j}$ = 1 iff word i in 1st sentence is aligned to word j in 2[nd]
- Features vector decompose over each aligned edge
  - $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{y_{i,j}=1} \mathbf{g}(\mathbf{x}, i,j)$
  - $\mathbf{g}(\mathbf{x}, i,j)$: various properties comparing i-th and j-th word
    - Difference in the position of the two words
    - Is part of speech of the two words the same?
- MAP: Maximum weight matching

Image from :
http://gate.ac.uk/sale/tao/alignment-editor.png

Ranking of search results in IR

- Input **x**: Query q, List of documents d_1 , d_2 ,…, d_n
- Output **y**:
  - Ranking of documents so that relevant documents appear before irrelevant ones
  - y_i = position of document d_i
- Feature vector **f**(**x, y**) defined as follows
  - **g**(d_i, q) = vector of properties relating d_i to q
    - Jaccard similarity between query words and document
    - Popularity of document d_i
  - $f(\mathbf{x}, \mathbf{y}) = \sum_{y_i < y_j} (\mathbf{g}(d_i, q) - \mathbf{g}(d_j, q))$
- MAP: rank documents on **w**.**g**(d_i, q)

Markov models (CRFs)

- Application: Image segmentation and many others
- **y** is a vector $y_1, y_2, .., y_n$ of discrete labels
- Features decompose over cliques of a triangulated graph
- MAP inference algorithms for graphical models, extensively researched
  - Junction trees for exact, many approximate algorithms
    - Special case: Viterbi
- Framework of structured models subsumes graphical models

Structured model

- Score of a prediction **y** for input **x**:
  - $s(\mathbf{x},\mathbf{y}) = \mathbf{w}.\ \mathbf{f}(\mathbf{x},\mathbf{y})$
- Prediction problem: find highest scoring output
  - $\mathbf{y}_* = \text{argmax}_{\mathbf{y}}\ s(\mathbf{x},\mathbf{y})$
  - Space of possible **y** exponentially large
  - Exploit decomposability of feature functions
    - $\mathbf{f}(\mathbf{x},\mathbf{y}) = \sum_c \mathbf{f}\,(\mathbf{x},\mathbf{y}_c,c)$
- Training problem: find **w** given many correct input-output pairs $(\mathbf{x}_1\ \ \mathbf{y}_1), (\mathbf{x}_2\ \ \mathbf{y}_2), \ldots, (\mathbf{x}_N\ \ \mathbf{y}_N)$

Max-margin loss surrogates

True error $E_i(\text{argmax}_{\mathbf{y}} \mathbf{w}.\mathbf{f}(\mathbf{x_i}, \mathbf{y}))$

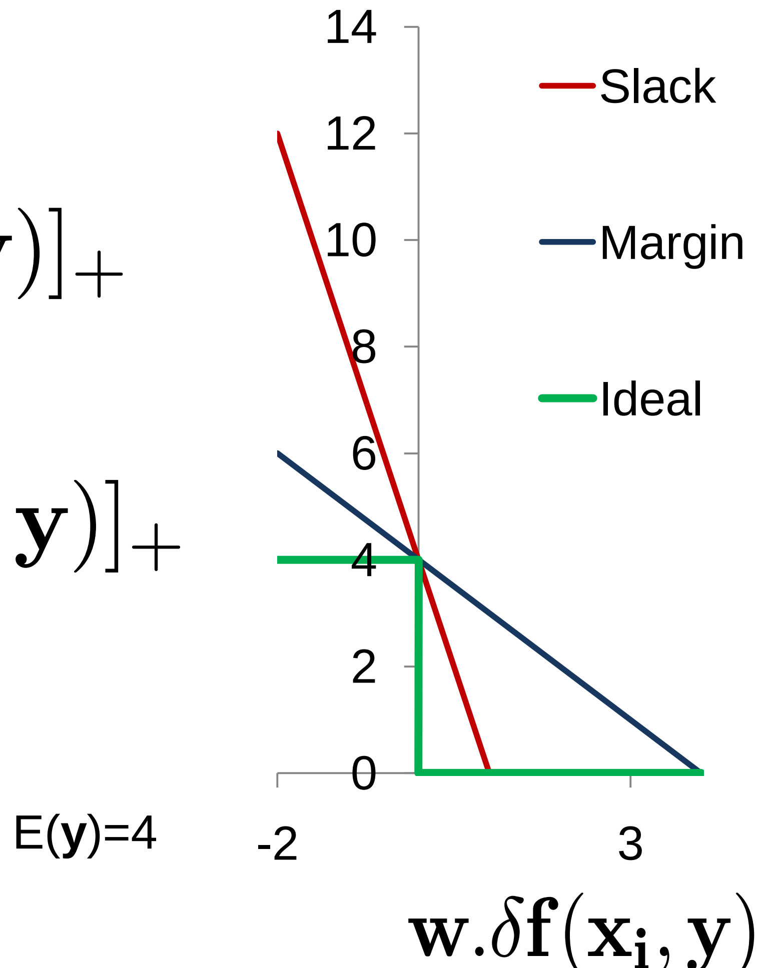
Let $\mathbf{w}.\delta\mathbf{f}(\mathbf{x_i}, \mathbf{y}) = \mathbf{w}.\mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) - \mathbf{w}.\mathbf{f}(\mathbf{x_i}, \mathbf{y})$

1. Margin Loss

$$\max_{\mathbf{y}} [E_i(\mathbf{y}) - \mathbf{w}.\delta\mathbf{f}(\mathbf{x_i}, \mathbf{y})]_+$$

2. Slack Loss

$$\max_{\mathbf{y}} E_i(\mathbf{y})[1 - \mathbf{w}.\delta\mathbf{f}(\mathbf{x_i}, \mathbf{y})]_+$$



Final optimization

- Margin

$$\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^{\mathbf{2}} + \mathbf{C}\sum_{\mathbf{i=1}}^{\mathbf{N}} \xi_{\mathbf{i}}$$

$$\text{s.t. } \mathbf{w}.\delta\mathbf{f_i}(\mathbf{y}) \geq \mathbf{E_i}(\mathbf{y}) - \xi_{\mathbf{i}} \quad \forall \mathbf{y} \neq \mathbf{y_i}, \mathbf{i} : \mathbf{1}\ldots\mathbf{N}$$

$$\xi_i \geq 0 \quad i : 1 \ldots N$$

- Slack

$$\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^{\mathbf{2}} + \mathbf{C}\sum_{\mathbf{i=1}}^{\mathbf{N}} \xi_{\mathbf{i}}$$

$$\text{s.t. } \mathbf{w}.\delta\mathbf{f_i}(\mathbf{y}) \geq \mathbf{1} - \frac{\xi_{\mathbf{i}}}{\mathbf{E_i}(\mathbf{y})} \quad \forall \mathbf{y} \neq \mathbf{y_i}, \mathbf{i} : \mathbf{1}\ldots\mathbf{N}$$

$$\xi_i \geq 0 \quad i : 1 \ldots N$$

Exponential number of constraints ➔ Use cutting plane

Margin Vs Slack

- Margin
  - Easy inference of most violated constraint for decomposable **f** and E

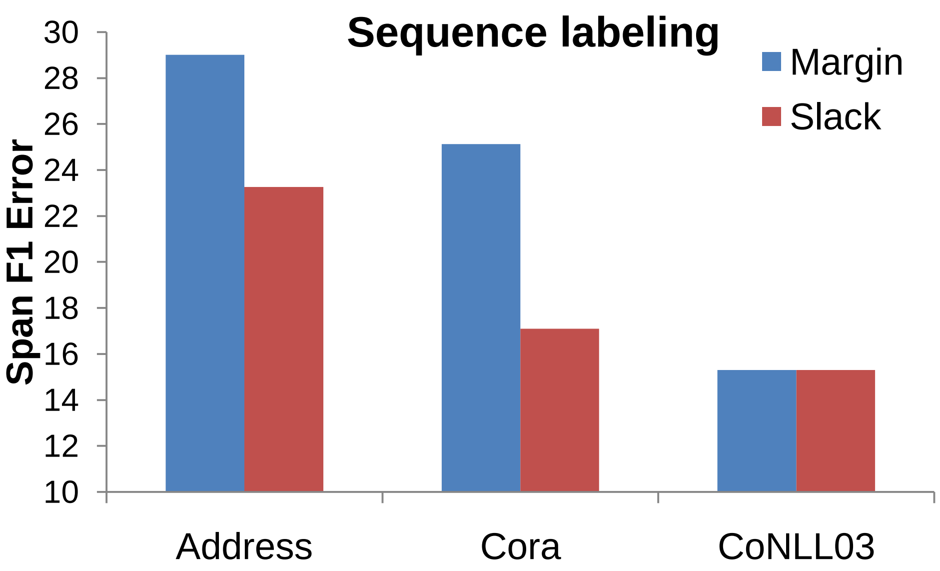
$$\mathbf{y}^M = \mathrm{argmax}_{\mathbf{y}}(\mathbf{w}.\mathbf{f}(\mathbf{x_i}, \mathbf{y}) + \mathbf{E_i}(\mathbf{y}))$$

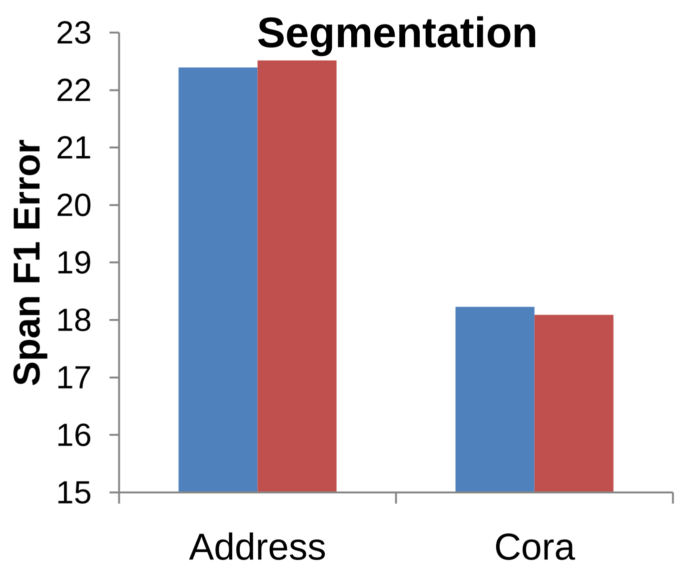
  - Too much importance to **y** far from margin
- Slack
  - Difficult inference of violated constraint

$$\mathbf{y}^S = \mathrm{argmax}_{\mathbf{y}}(\mathbf{w}.\mathbf{f_i}(\mathbf{y}) - \frac{\xi_{\mathbf{i}}}{\mathbf{E_i}(\mathbf{y})})$$

  - Zero loss of everything outside margin
    - Higher accuracy.

Accuracy of Margin vs Slack





Slack scaling up to 25% better than Margin scaling.

Approximating Slack inference

- Slack inference: $\max_{\mathbf{y}} s(\mathbf{y}) - \xi / E(\mathbf{y})$
  - Decomposability of E(**y**) cannot be exploited.

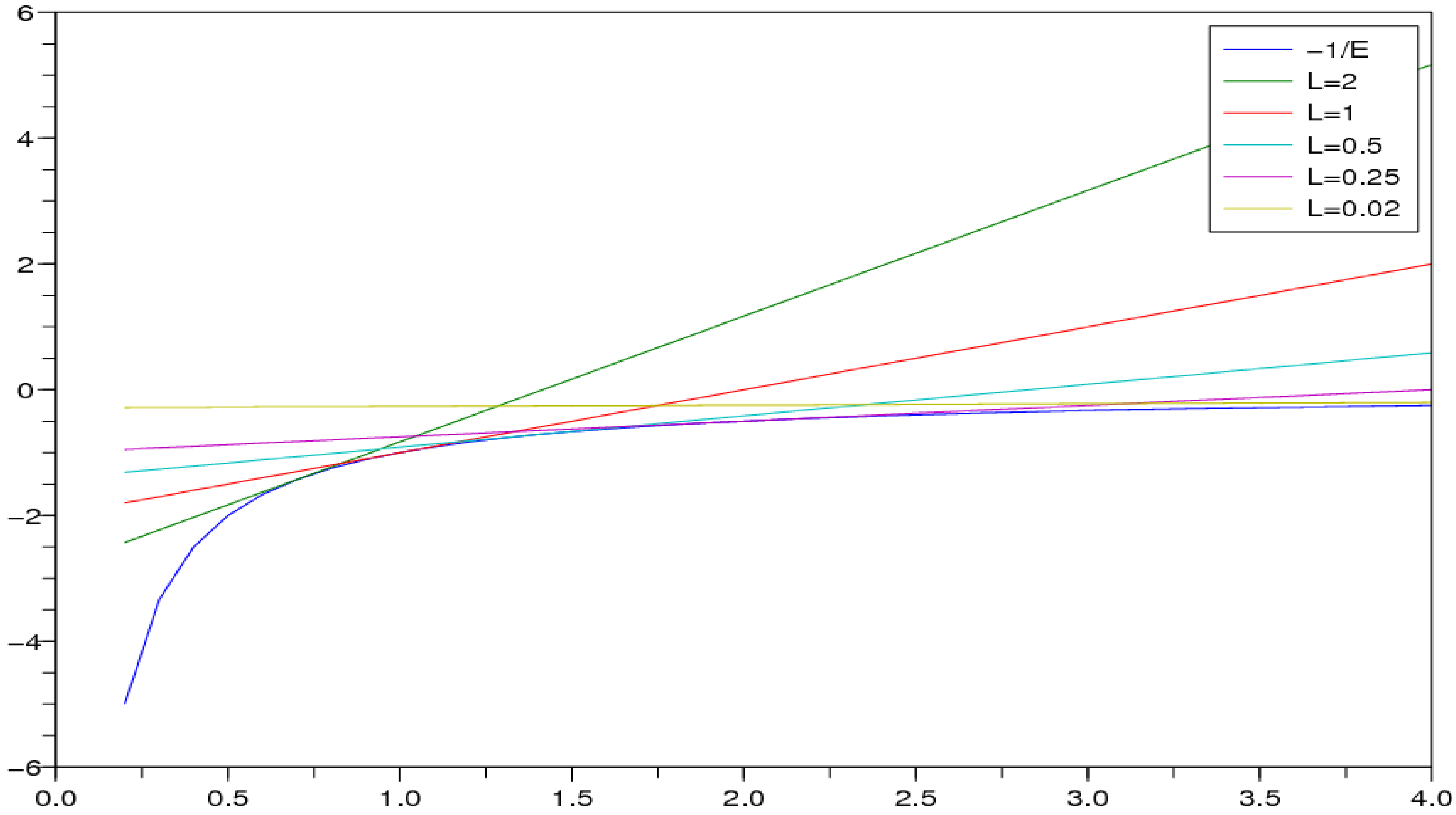
- $-\xi/E(\mathbf{y})$ is concave in E(**y**)
- Variational method to rewrite as linear function

$$-\frac{\xi}{E(\mathbf{y})} = \min_{\lambda \geq 0} \lambda E(\mathbf{y}) - 2\sqrt{(}\xi\lambda)$$

Approximating slack inference

- s(**y**)-ξ/E(**y**) is concave in E(**y**)
- Its variational form.

$$s(\mathbf{y}) - \frac{\xi}{E(\mathbf{y})} = \min_\lambda s(\mathbf{y}) + \lambda E(\mathbf{y}) - 2\sqrt{(}\xi\lambda)$$



Approximating Slack inference

- Now approximate the inference problem as:

$$\max_{\mathbf{y}} \left(s(\mathbf{y}) - \frac{\xi}{E(\mathbf{y})} \right) \quad = \quad \max_{\mathbf{y}} \min_{\lambda \geq 0} s(\mathbf{y}) + \lambda E(\mathbf{y}) - 2\sqrt{\xi\lambda}$$

$$\cdot \quad \min_{\lambda \geq 0} \boxed{\max_{\mathbf{y}} s(\mathbf{y}) + \lambda E(\mathbf{y})} - 2\sqrt{\xi\lambda}$$

Same tractable MAP as in Margin Scaling

Approximating slack inference

- Now approximate the inference problem as:

$$\max_{\mathbf{y}} \left(s(\mathbf{y}) - \frac{\xi}{E(\mathbf{y})} \right) \quad = \quad \max_{\mathbf{y}} \min_{\lambda \geq 0} s(\mathbf{y}) + \lambda E(\mathbf{y}) - 2\sqrt{\xi\lambda}$$

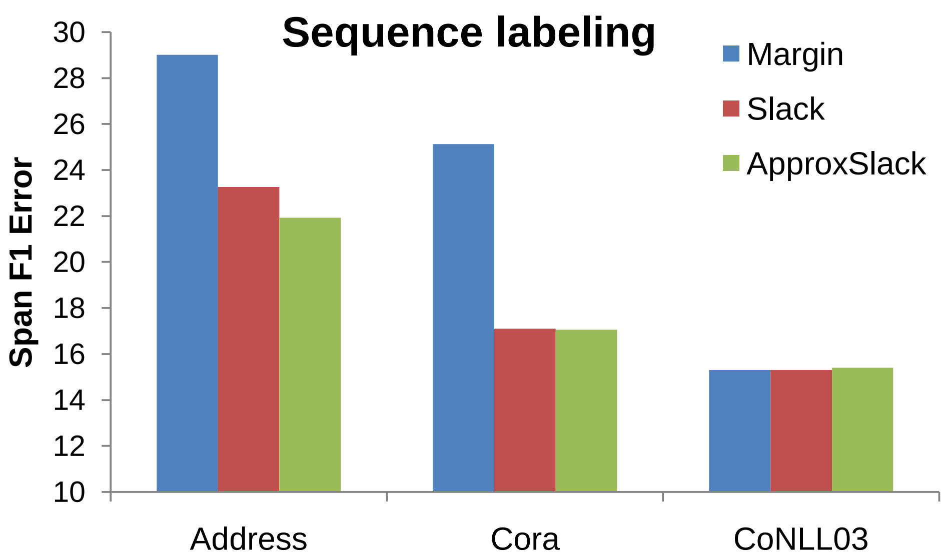
$$\cdot \quad \min_{\lambda \geq 0} \boxed{\max_{\mathbf{y}} s(\mathbf{y}) + \lambda E(\mathbf{y}) - 2\sqrt{\xi\lambda}}$$

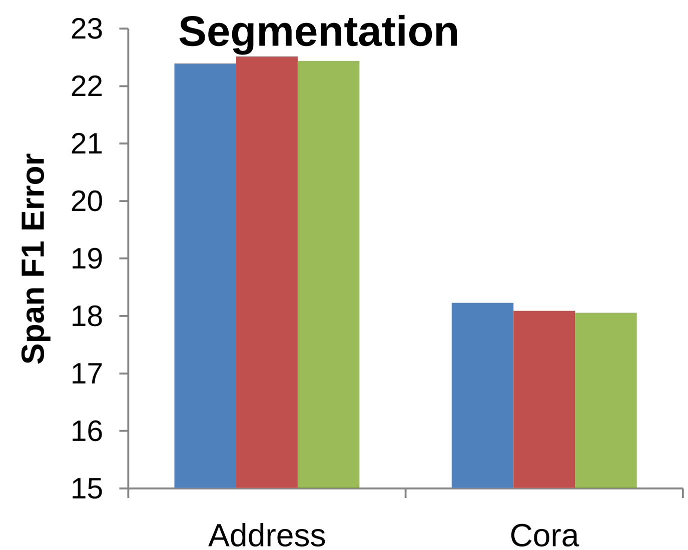
Same tractable MAP as in margin scaling

Convex in λ ➔ minimize using line search,

Bounded interval $[\lambda_l, \lambda_u]$ exists since only want violating y.

Slack Vs ApproxSlack

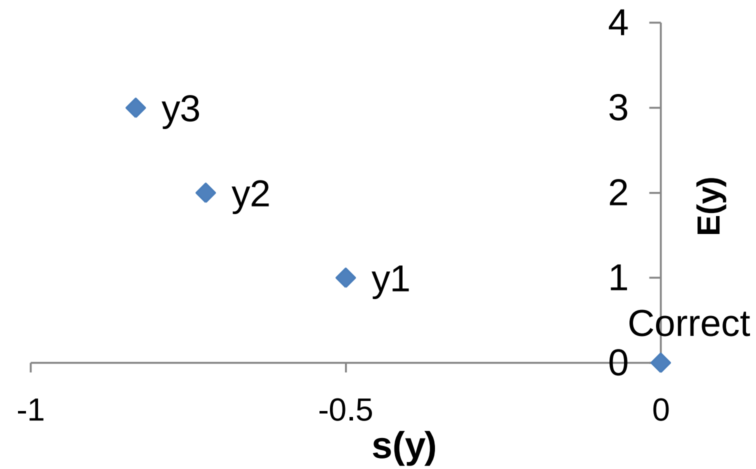




ApproxSlack gives the accuracy gains of Slack scaling while requiring same the MAP inference same as Margin scaling.

Limitation of ApproxSlack

- Cannot ensure that a violating **y** will be found even if it exists
  - No λ can ensure that.
- Proof:
  - $s(y_1)$=-1/2 $E(y_1) = 1$
  - $s(y_2) = -13/18$ $E(y_2) = 2$
  - $s(y_3) = -5/6$ $E(y_3) = 3$
  - s(correct) = 0
  - $\xi = 19/36$
  - y_2 has highest s(y)-ξ/E(y) and is violating.
  - No λ can score y_2 higher than both y_1 and y_2



Max-margin formulations

- Margin scaling

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

$$\text{s.t. } \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}_\mathrm{i}) \geq \mathrm{E}_\mathrm{i}(\mathbf{y}) + \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}) - \xi_\mathrm{i} \quad \forall \mathbf{y} \neq \mathbf{y}_\mathrm{i}, \forall \mathrm{i}$$

$$\xi_i \geq 0 \quad \forall i$$

- Slack scaling

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

$$\text{s.t. } \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}_\mathrm{i}) \geq 1 + \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}) - \frac{\xi_\mathrm{i}}{\mathrm{E}_\mathrm{i}(\mathbf{y})} \quad \forall \mathbf{y} \neq \mathbf{y}_\mathrm{i}, \forall \mathrm{i}$$

$$\xi_i \geq 0 \quad \forall i$$

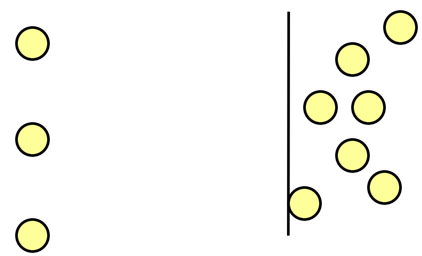
The pitfalls of a single shared slack variables

- Inadequate coverage for decomposable losses

Non-separable: $y_2 = [0\ 0\ 1]$

Separable: $y_1 = [0\ 1\ 0]$

Correct : $y_0 = [0\ 0\ 0]$



s=0 s=-3

Margin/Slack loss = 1.

Since y_2 non-separable from y_0, ξ=1, Terminate.

Premature since different features may be involved.

A new loss function: PosLearn

- Ensure margin at each loss position

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\sum_c \xi_{i,c}$$

$$\text{s.t} \quad \mathbf{w}.\mathbf{f}(\mathbf{x}_i, \mathbf{y}_i) \geq 1 + \mathbf{w}.\mathbf{f}(\mathbf{x}_i, \mathbf{y}) - \frac{\xi_{i,c}}{E_{i,c}(\mathbf{y}_c)} \quad \forall \mathbf{y} : \mathbf{y}_c \neq \mathbf{y}_{i,c}$$

$$\xi_{i,c} \geq 0 \quad i : 1 \ldots N, \forall c$$

- Compare with slack scaling.

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

$$\text{s.t. } \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}_\mathrm{i}) \geq 1 + \mathbf{w}.\mathbf{f}(\mathbf{x}_\mathrm{i}, \mathbf{y}) - \frac{\xi_\mathrm{i}}{\mathrm{E}_\mathrm{i}(\mathbf{y})} \quad \forall \mathbf{y} \neq \mathbf{y}_\mathrm{i}, \forall \mathrm{i}$$

$$\xi_i \geq 0 \quad \forall i$$

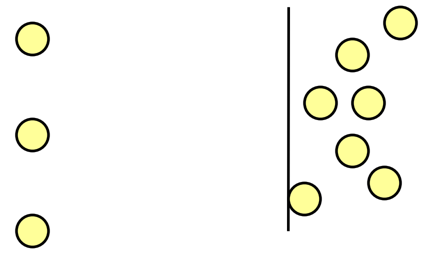
The pitfalls of a single shared slack variables

- Inadequate coverage for decomposable losses

Non-separable: $y_{2} =$ [0 0 1]

Separable: $y_{1} =$ [0 1 0]

Correct : $y_{0} =$ [0 0 0]

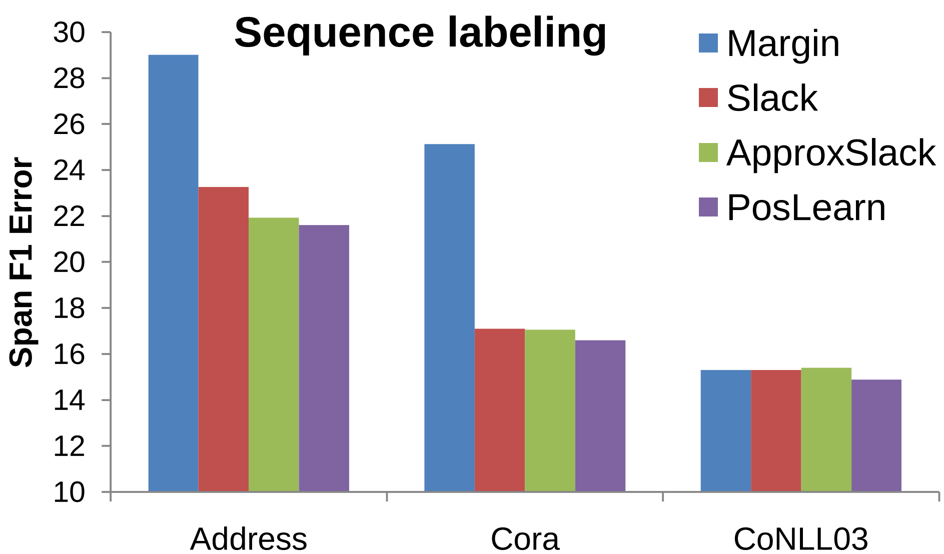


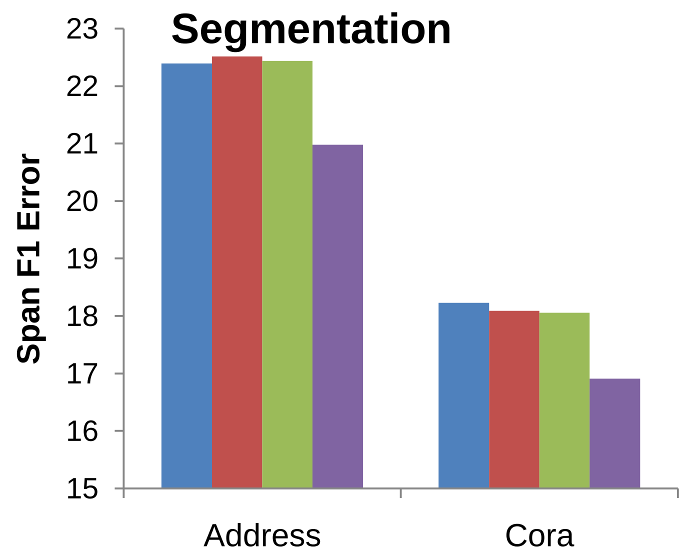
s=0 s=-3

Margin/Slack loss = 1.
Since y_2 non-separable from y_0, ξ=1, Terminate.
Premature since different features may be involved.

PosLearn loss = 2
Will continue to optimize for y_1 even after slack ξ_3 becomes 1

Comparing loss functions





PosLearn: same or better than Slack and ApproxSlack

Inference for PosLearn QP

- Cutting plane inference
  - For each position c, find best **y** that is wrong at c

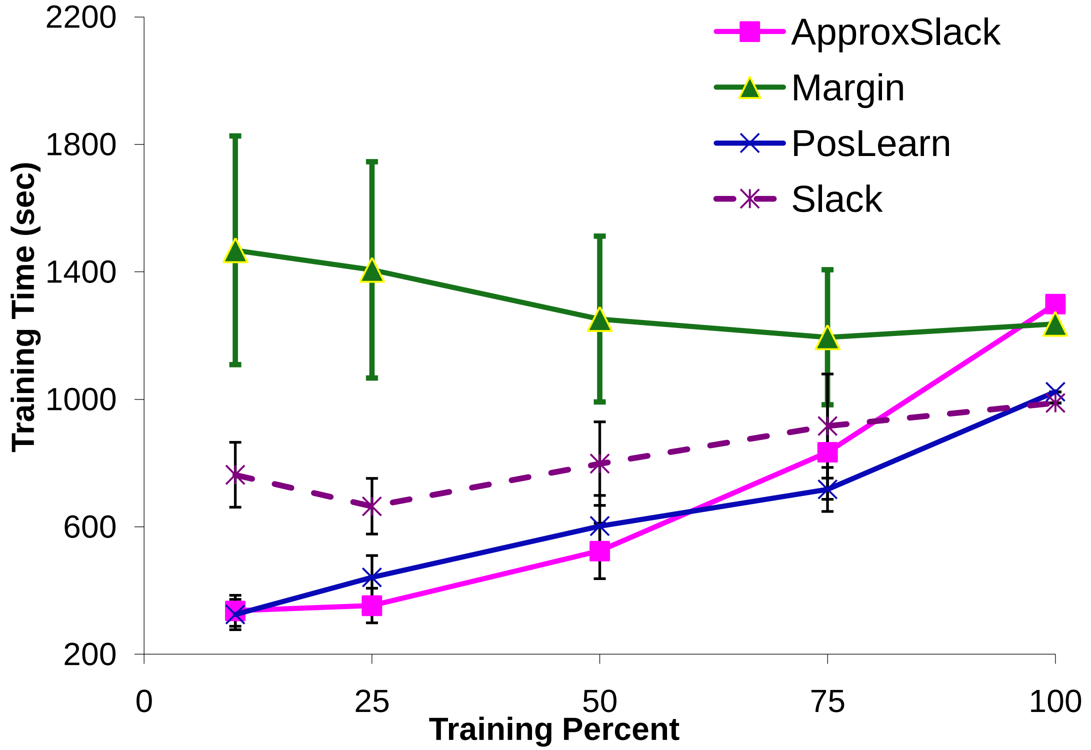
$$\max_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} \left(s_i(\mathbf{y}) - \frac{\xi_{i,c}}{E_{i,c}(\mathbf{y}_c)} \right) = \max_{\mathbf{y}_c \neq \mathbf{y}_{i,c}} \left(\boxed{\max_{\mathbf{y} \sim \mathbf{y}_c} s_i(\mathbf{y})} - \frac{\xi_{i,c}}{E_{i,c}(\mathbf{y}_c)} \right)$$

Small enumerable set

MAP with restriction, easy!

- Solve simultaneously for all positions c
  - Markov models: Max-Marginals
  - Segmentation models: forward-backward passes
  - Parse trees

Running time



Margin scaling might take time with less data since good constraints may not be found early

PosLearn adds more constraints but needs fewer iterations.

Summary of training

1. Margin scaling popular due to computational reasons, but slack scaling more accurate
   - A variational approximation for slack inference

2. Single slack variable inadequate for structured models where errors are additive
   - A new loss function that ensures margin at each possible error position of **y**