Machine Learning Department
School of Computer Science
Carnegie Mellon University

# MAP Inference with MILP

Matt Gormley
Lecture 13
Oct. 9, 2019

# Q&A

**Q:** What is the “Study on Supporting and Improving Teaching at the University Level” mentioned on Piazza?

**A:** …

# Q&A

**Q:** Do we **really** have to write a project report and create a video presentation?

**A:** Nope! Not anymore. We've dramatically improved the schedule for the rest of the semester. Here are the highlights:

- 10-418/618 students:
  - **Final Exam:** Thu, Dec-05 in the evening (last week of classes)
- 10-618 students:
  - **Midway Poster Session**: Mon, Nov-25 (date/time TBD)
  - **Final Poster Session**: during final exam week (Dec 9 – 15, date/time TBD)
  - The two posters (midway poster, final poster) replace the old report/video milestones

# Reminders

- **Homework 2: BP for Syntax Trees**
  - Out: Sat, Sep. 28
  - Due: Sat, Oct. 12 at 11:59pm
- **Last chance to switch between 10-418 / 10-618 is October 7th (drop deadline)**
- **Today's after-clas office hours are un-cancelled (i.e. I am having them)**

# LINEAR PROGRAMMING & INTEGER LINEAR PROGRAMMING

# Integer Linear Programming

***Whiteboard***

– Branch and bound for an ILP in 2D

# Branch and Bound

**Algorithm 2.1** Branch-and-bound

*Input*: Minimization problem instance $R$.

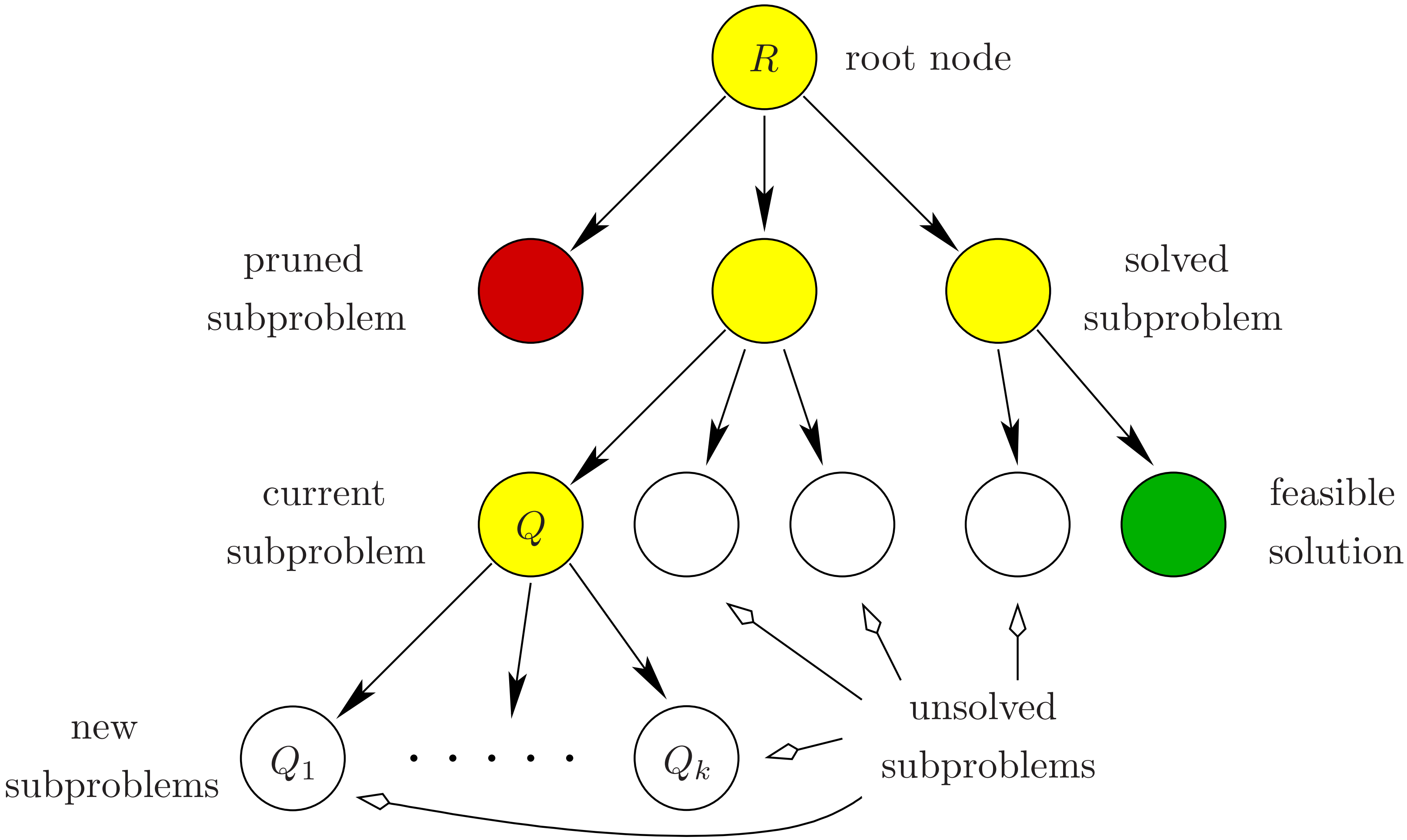
*Output*: Optimal solution $x^\star$ with value $c^\star$, or conclusion that $R$ has no solution, indicated by $c^\star = \infty$.

1. Initialize $\mathcal{L} := \{R\}$, $\hat{c} := \infty$. [*init*]
2. If $\mathcal{L} = \emptyset$, stop and return $x^\star = \hat{x}$ and $c^\star = \hat{c}$. [*abort*]
3. Choose $Q \in \mathcal{L}$, and set $\mathcal{L} := \mathcal{L} \setminus \{Q\}$. [*select*]
4. Solve a relaxation $Q_{\text{relax}}$ of $Q$. If $Q_{\text{relax}}$ is empty, set $\check{c} := \infty$. Otherwise, let $\check{x}$ be an optimal solution of $Q_{\text{relax}}$ and $\check{c}$ its objective value. [*solve*]
5. If $\check{c} \geq \hat{c}$, goto Step 2. [*bound*]
6. If $\check{x}$ is feasible for $R$, set $\hat{x} := \check{x}$, $\hat{c} := \check{c}$, and goto Step 2. [*check*]
7. Split $Q$ into subproblems $Q = Q_1 \cup \ldots \cup Q_k$, set $\mathcal{L} := \mathcal{L} \cup \{Q_1, \ldots, Q_k\}$, and goto Step 2. [*branch*]



Slide from Achterberg (thesis, 2007)

# Branch and Bound




Slide from Achterberg (thesis, 2007)

# Branch and Bound

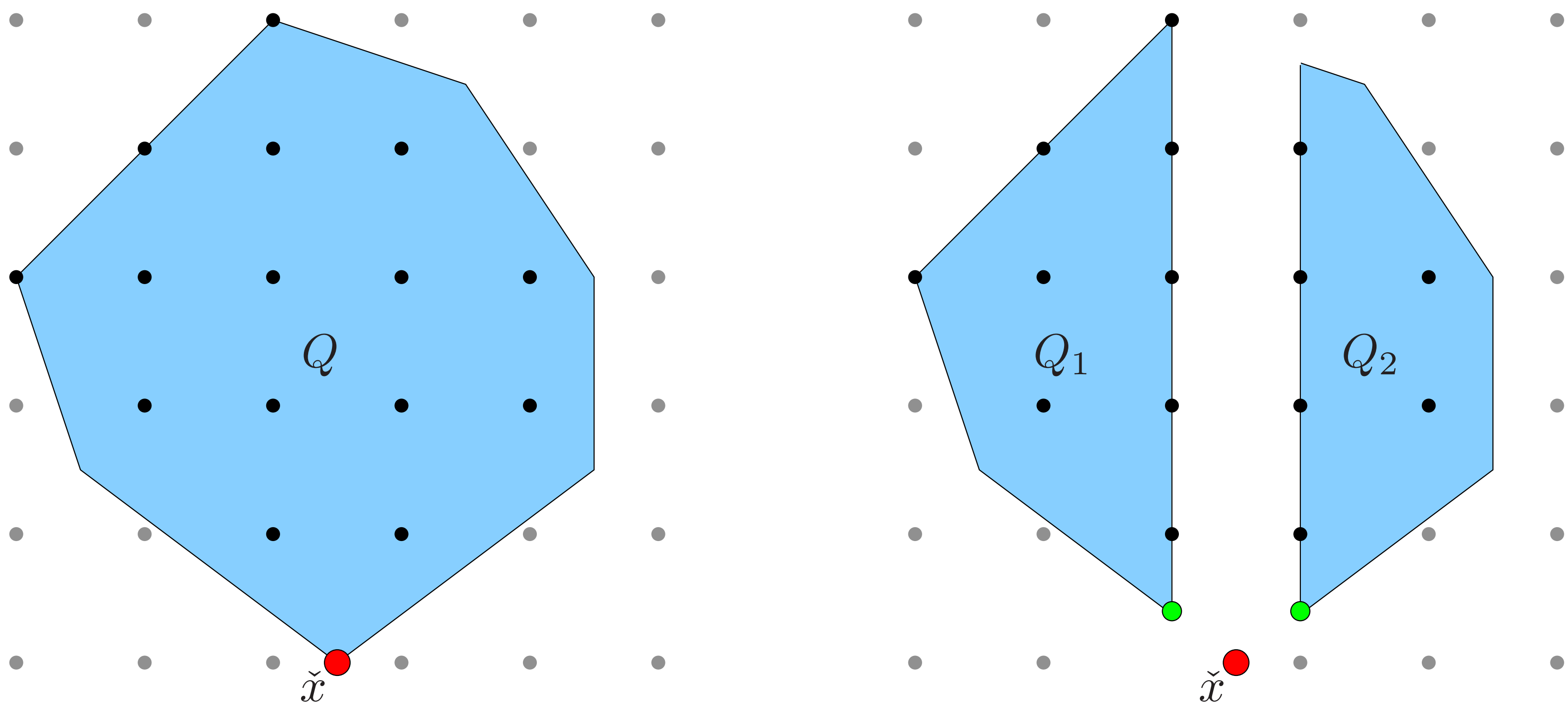



**Figure 2.2.** LP based branching on a single fractional variable.



Slide from Achterberg (thesis, 2007)

# MAP INFERENCE AS MATHEMATICAL PROGRAMMING

# Exact Inference

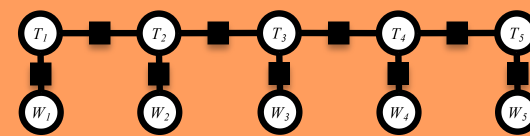
Recall...

## 1. Data

$$\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$$


## 2. Model

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$




## 3. Objective

$$\ell(\theta; \mathcal{D}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}^{(n)} \mid \boldsymbol{\theta})$$

## 5. Inference

**1. Marginal Inference**

$$p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

**2. Partition Function**

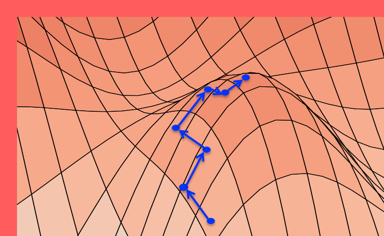
$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

**3. MAP Inference**

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

## 4. Learning

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathcal{D})$$



# 5. Inference

Recall...

Three Tasks: **(All three are NP-Hard in the general case)**

**1. Marginal Inference**
Compute marginals of variables and cliques

$$p(x_i) = \sum_{\boldsymbol{x}' : x'_i = x_i} p(\boldsymbol{x}' \mid \boldsymbol{\theta}) \quad \Bigg| \quad p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

**2. Partition Function**
Compute the normalization constant

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

**3. MAP Inference**
Compute variable assignment with highest probability

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Recall…

# 5. Inference

Three Tasks:

**1. Marginal Inference**
Compute marginals of variables and cliques

$$p(x_i) = \sum_{\boldsymbol{x}' : x'_i = x_i} p(\boldsymbol{x}' \mid \boldsymbol{\theta}) \quad \Bigg| \quad p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

**2. Partition Function**
Compute the normalization constant

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

**3. MAP Inference (NP-Hard in the general case)**
Compute variable assignment with highest probability

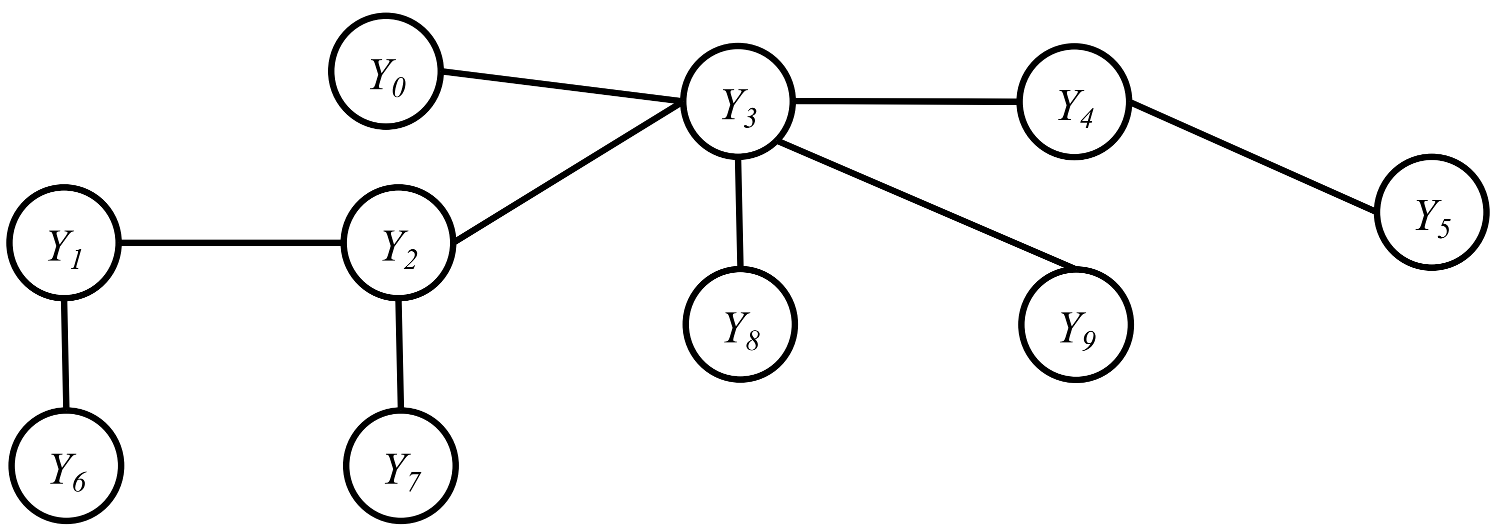
$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

# MAP Inference

Suppose we want to predict the highest likelihood structure y, given observations x and parameters w.

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \log p_w(y|x)$$

$$= \underset{\mathbf{y}}{\operatorname{argmax}} \sum_j \mathbf{w}^T f_{\text{node}}(x_j, y_j) + \sum_{j,k} \mathbf{w}^T f_{\text{edge}}(\mathbf{x}_{jk}, y_j, y_k)$$



# MAP Inference

Suppose we want to predict the highest likelihood structure y, given observations x and parameters w.

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \log p_w(y|x)$$

$$= \underset{\mathbf{y}}{\operatorname{argmax}} \sum_j \mathbf{w}^T f_{\text{node}}(x_j, y_j) + \sum_{j,k} \mathbf{w}^T f_{\text{edge}}(\mathbf{x}_{jk}, y_j, y_k)$$

**Idea:**

1. Reformulate the problem as an integer linear program (ILP) – note that this is just going to be a new way of writing down the problem: y → z
2. Then remove the integer constraints (i.e. solve the linear program (LP) relaxation)

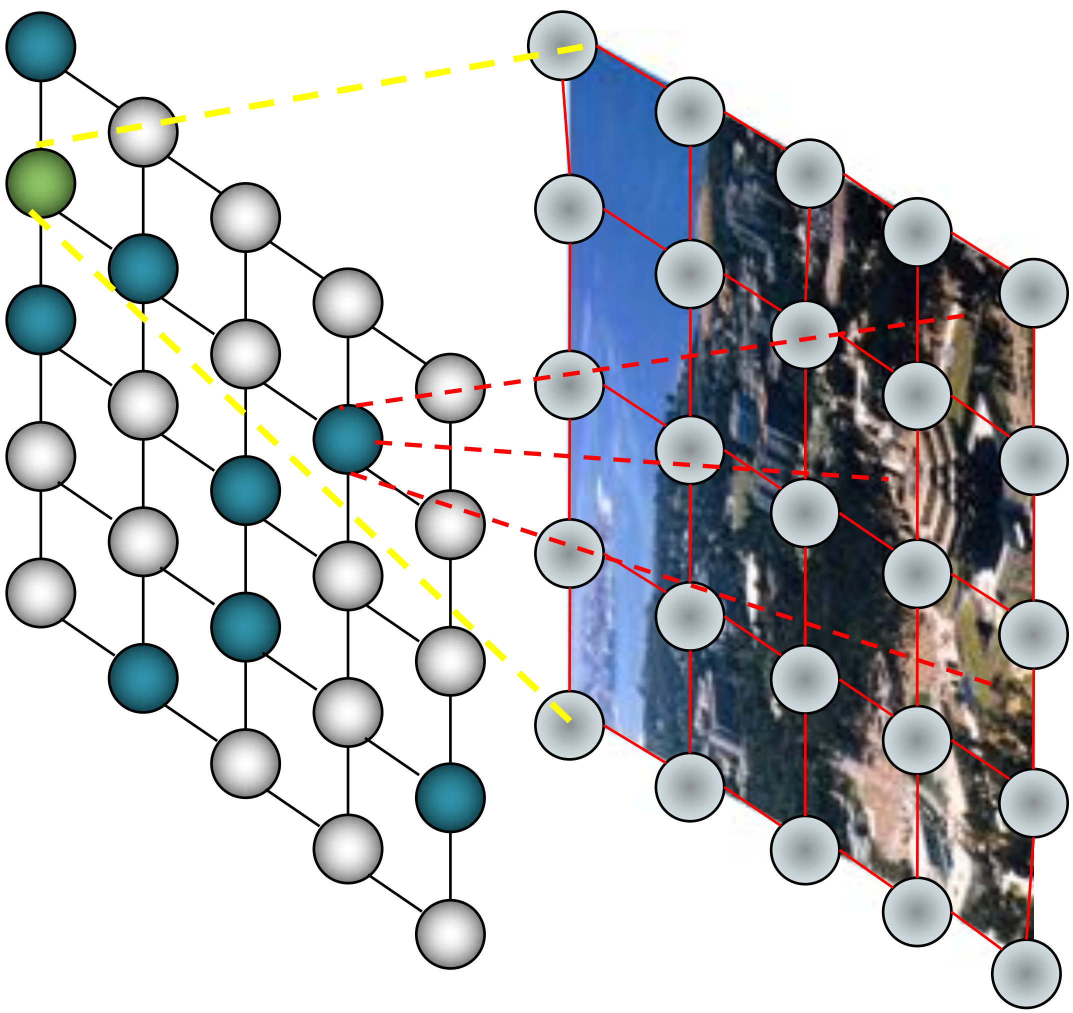
**Lemma:** (Wainwright et al., 2002) If there is a unique MAP assignment, the LP relaxation of the ILP above is guaranteed to have an integer solution, which is exactly the MAP solution!

# Integer Linear Programming

***Whiteboard***

- MAP Inference for a Binary Pairwise MRF as an ILP
- Question: What if we have non-binary variables?

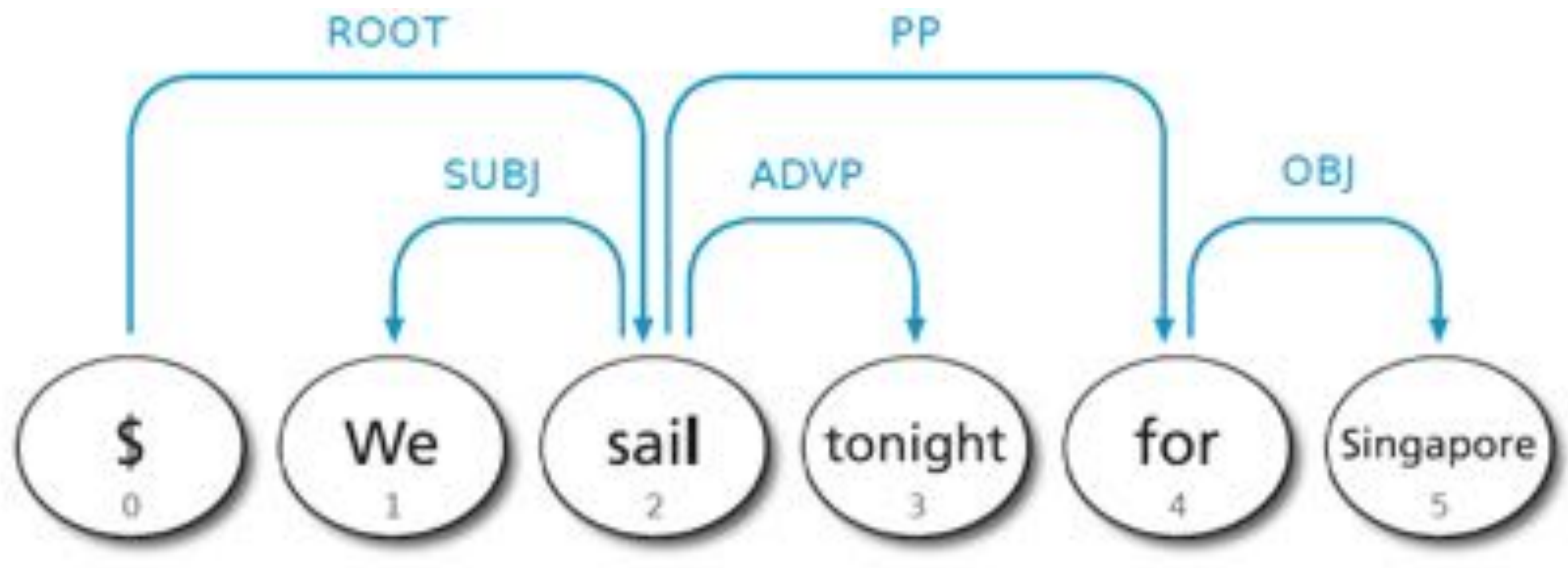
# Image Segmentation



$$p_\theta(y \mid x) = \frac{1}{Z(\theta, x)} \exp\left\{\sum_c \theta_c f_c(x, y_c)\right\}$$

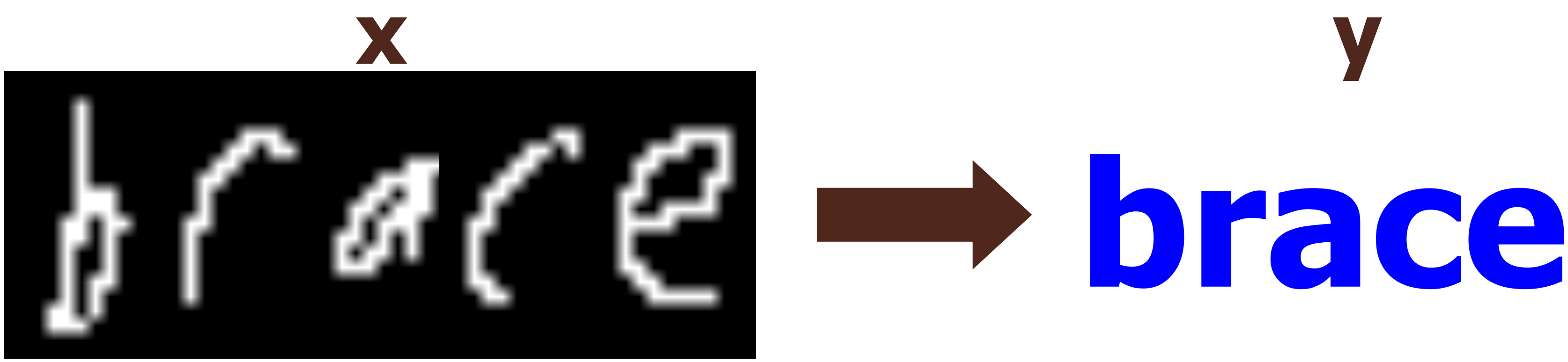
- Jointly segmenting/annotating images
- Image-image matching, image-text matching
- Problem:
  - Given structure (feature), learning $\vec{\theta}$
  - Learning sparse, interpretable, **predictive** structures/features

# Dependency parsing of Sentences

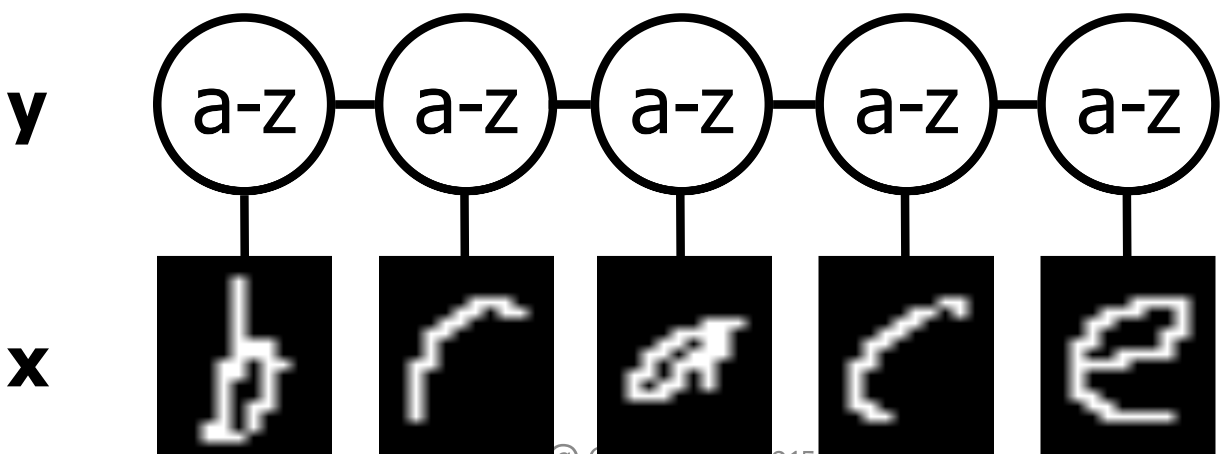



Challenge:
Structured outputs, and globally constrained to be a valid tree

# OCR example




Sequential structure

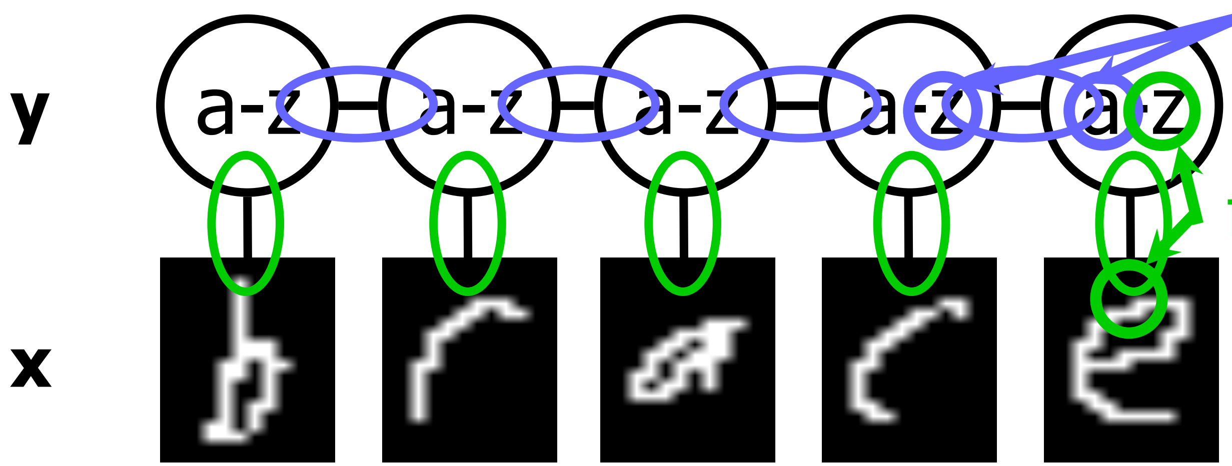


# Linear-chain CRF for OCR

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \phi(\mathbf{x}_i, y_i) \prod_i \phi(y_i, y_{i+1})$$

$$\phi(\mathbf{x}_i, y_i) = \exp\{\sum_\alpha w_\alpha f_\alpha(\mathbf{x}_i, y_i)\}$$

$$\phi(y_i, y_{i+1}) = \exp\{\sum_\beta w_\beta f_\beta (y_i, y_{i+1})\}$$




$f_\beta(y,y') = I(y=\text{'z'}, y'=\text{'a'})$

$f_\alpha(\mathbf{x},y) = I(x_p=1, y=\text{'z'})$

*Lafferty et al. 01

# $y \Rightarrow z$ map for linear chain structures

OCR example: $\mathbf{y} =$ 'ABABB';

$\mathbf{z}$'s are the indicator variables for the corresponding classes (alphabet)

| | $z_1(m)$ | $z_2(m)$ | $z_3(m)$ | $z_4(m)$ | $z_5(m)$ |
|---|---|---|---|---|---|
| A | 1 | 0 | 1 | 0 | 0 |
| B | 0 | 1 | 0 | 1 | 1 |
| : | : | : | : | : | : |
| Z | 0 | 0 | 0 | 0 | 0 |

$z_{12}(m,n)$

| | A | B | . | Z |
|---|---|---|---|---|
| A | 0 | 1 | . | 0 |
| B | 0 | 0 | . | 0 |
| : | . | . | . | 0 |
| Z | 0 | 0 | 0 | 0 |

$z_{23}(m,n)$

| | A | B | . | Z |
|---|---|---|---|---|
| A | 0 | 0 | . | 0 |
| B | 1 | 0 | . | 0 |
| : | . | . | . | 0 |
| Z | 0 | 0 | 0 | 0 |

$z_{34}(m,n)$

| | A | B | . | Z |
|---|---|---|---|---|
| A | 0 | 1 | . | 0 |
| B | 0 | 0 | . | 0 |
| : | . | . | . | 0 |
| Z | 0 | 0 | 0 | 0 |

$z_{45}(m,n)$

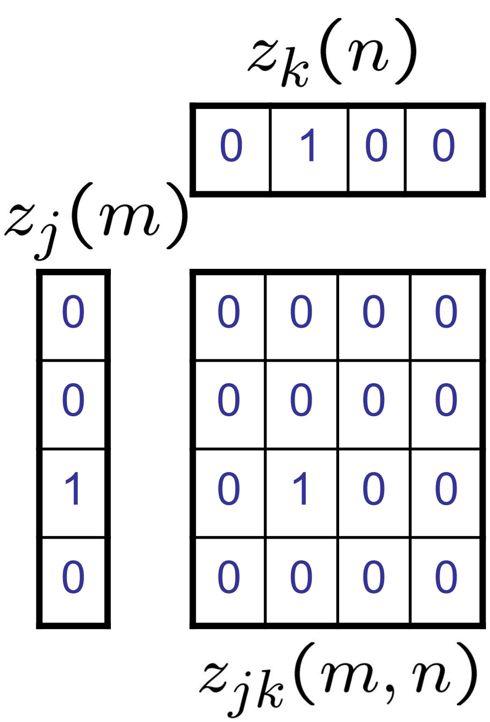
| | A | B | . | Z |
|---|---|---|---|---|
| A | 0 | 0 | . | 0 |
| B | 0 | 1 | . | 0 |
| : | . | . | . | 0 |
| Z | 0 | 0 | 0 | 0 |

# $y \Rightarrow z$ map for linear chain structures

$$\max_{\mathbf{y}} \sum_j \mathbf{w}^T f_{\text{node}}(x_j, y_j) + \sum_{j,k} \mathbf{w}^T f_{\text{edge}}(\mathbf{x}_{jk}, y_j, y_k)$$

Rewriting the maximization function in terms of indicator variables:

$$\max_{\mathbf{z}} \sum_{j,m} z_j(m) \left[\mathbf{w}^\top \mathbf{f}_{\text{node}}(\mathbf{x}_j, m)\right] + \sum_{jk,m,n} z_{jk}(m,n) \left[\mathbf{w}^\top \mathbf{f}_{\text{edge}}(\mathbf{x}_{jk}, m, n)\right]$$




$$z_j(m) \geq 0; \; z_{jk}(m,n) \geq 0;$$

normalization $\sum_m z_j(m) = 1$

agreement $\sum_n z_{jk}(m,n) = z_j(m)$

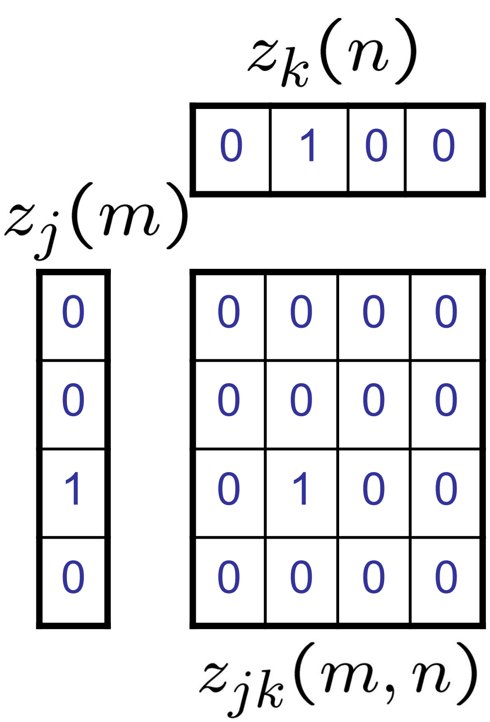
integer $z_j(m) \in \mathcal{Z}, \; z_{jk}(m,n) \in \mathcal{Z}$

# y ⇒ z map for linear chain structures

$$\max_{\mathbf{y}} \sum_j \mathbf{w}^T f_{\text{node}}(x_j, y_j) + \sum_{j,k} \mathbf{w}^T f_{\text{edge}}(\mathbf{x}_{jk}, y_j, y_k)$$

Rewriting the maximization function in terms of indicator variables:

$$\left.\begin{aligned} \max_{\mathbf{z}} \quad & \sum_{j,m} z_j(m) \left[ \mathbf{w}^\top \mathbf{f}_{\text{node}}(\mathbf{x}_j, m) \right] \\ + & \sum_{jk,m,n} z_{jk}(m,n) \left[ \mathbf{w}^\top \mathbf{f}_{\text{edge}}(\mathbf{x}_{jk}, m, n) \right] \end{aligned}\right\} (\mathbf{F}^\top \mathbf{w} \quad )^\top \mathbf{z}$$




$$z_j(m) \geq 0; \;\; z_{jk}(m,n) \geq 0;$$

normalization $\sum_m z_j(m) = 1$

agreement $\sum_n z_{jk}(m,n) = z_j(m)$

$$\Big\} \; \mathbf{A}\mathbf{z} = \mathbf{b}$$

$$\max_{\mathbf{A}\mathbf{z}=\mathbf{b}} \; (\mathbf{F}^\top \mathbf{w} \quad )^\top \mathbf{z}$$

# MAP Inference

Suppose we want to predict the highest likelihood structure y, given observations x and parameters w.

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \log p_w(y|x)$$
$$= \underset{\mathbf{y}}{\operatorname{argmax}} \sum_j \mathbf{w}^T f_{\text{node}}(x_j, y_j) + \sum_{j,k} \mathbf{w}^T f_{\text{edge}}(\mathbf{x}_{jk}, y_j, y_k)$$

**Idea:**

1. Reformulate the problem as an integer linear program (ILP) – note that this is just going to be a new way of writing down the problem: y → z
2. Then remove the integer constraints (i.e. solve the linear program (LP) relaxation)

**Lemma:** (Wainwright et al., 2002) If there is a unique MAP assignment, the LP relaxation of the ILP above is guaranteed to have an integer solution, which is exactly the MAP solution!

Looking ahead, we're going to use MAP inference as subroutine within Structured Perceptron and M3Ns (Structured SVM)

This is a preview of the results to come…

# MAP INFERENCE AND LEARNING

# Max (Conditional) Likelihood

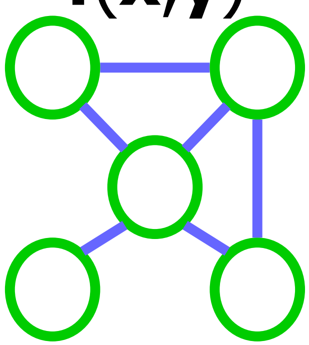
D

$\mathbf{x}^1, \mathbf{t}(\mathbf{x}^1)$

...

$\mathbf{x}^m, \mathbf{t}(\mathbf{x}^m)$

**f(x,y)**



**Estimation**

$$\text{maximize}_{\mathbf{w}} \sum_{\mathbf{x} \in D} \log P_{\mathbf{w}}(\mathbf{t}(\mathbf{x}) \mid \mathbf{x})$$

**Classification**

$$\arg\max_{\mathbf{y}} \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$$

$$\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) - \log Z_{\mathbf{w}}(\mathbf{x})$$

Don't need to learn entire distribution!

# Results: Handwriting Recognition

Length: ~8 chars

Letter: 16x8 pixels

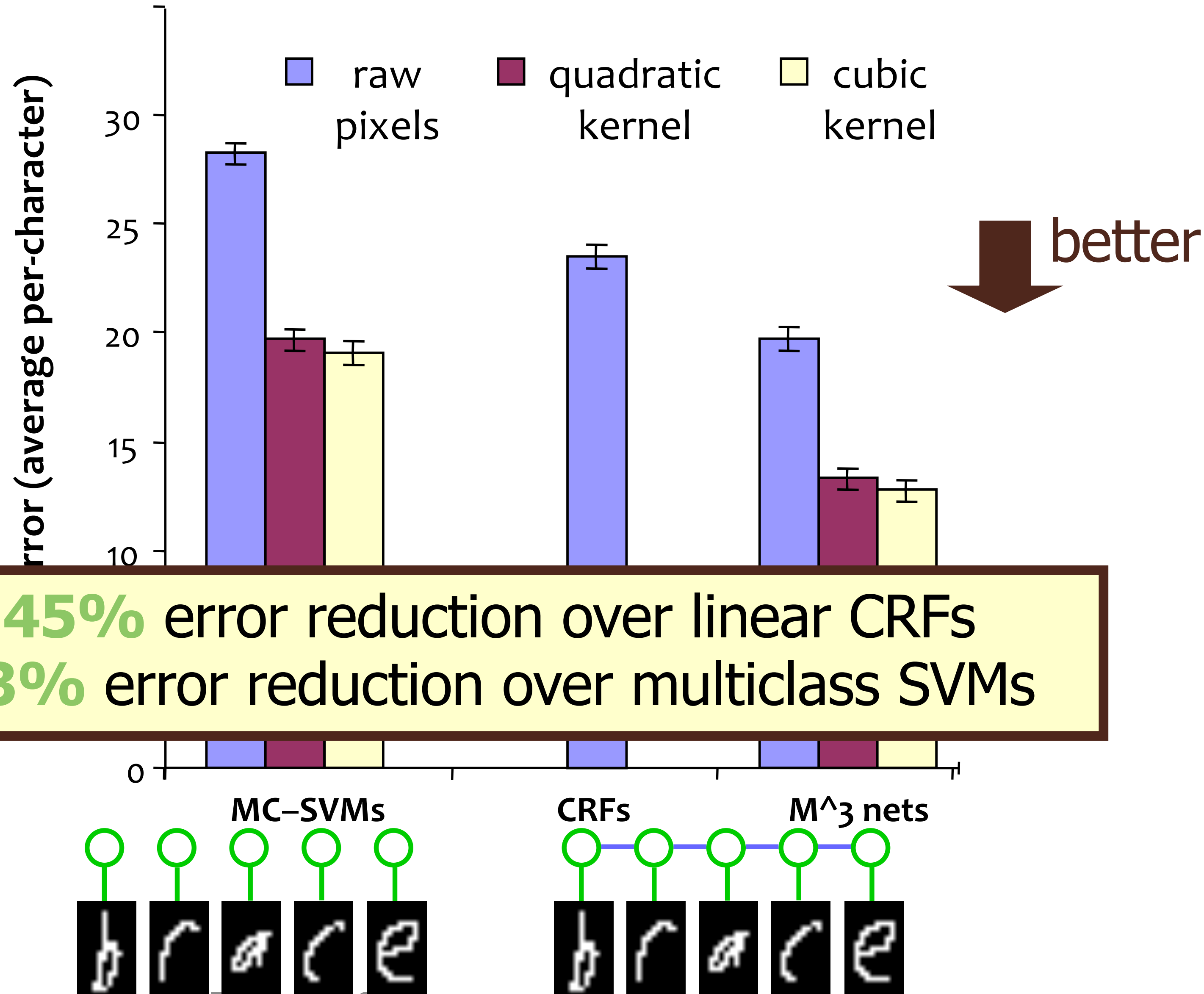
10-fold Train/Test

5000/50000 letters

600/6000 words

Models:

- Multiclass-SVMs*
- CRFs
- M³ nets



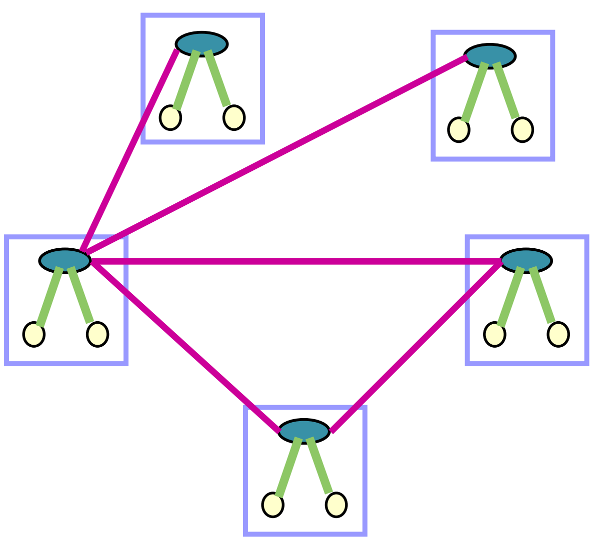

**45%** error reduction over linear CRFs

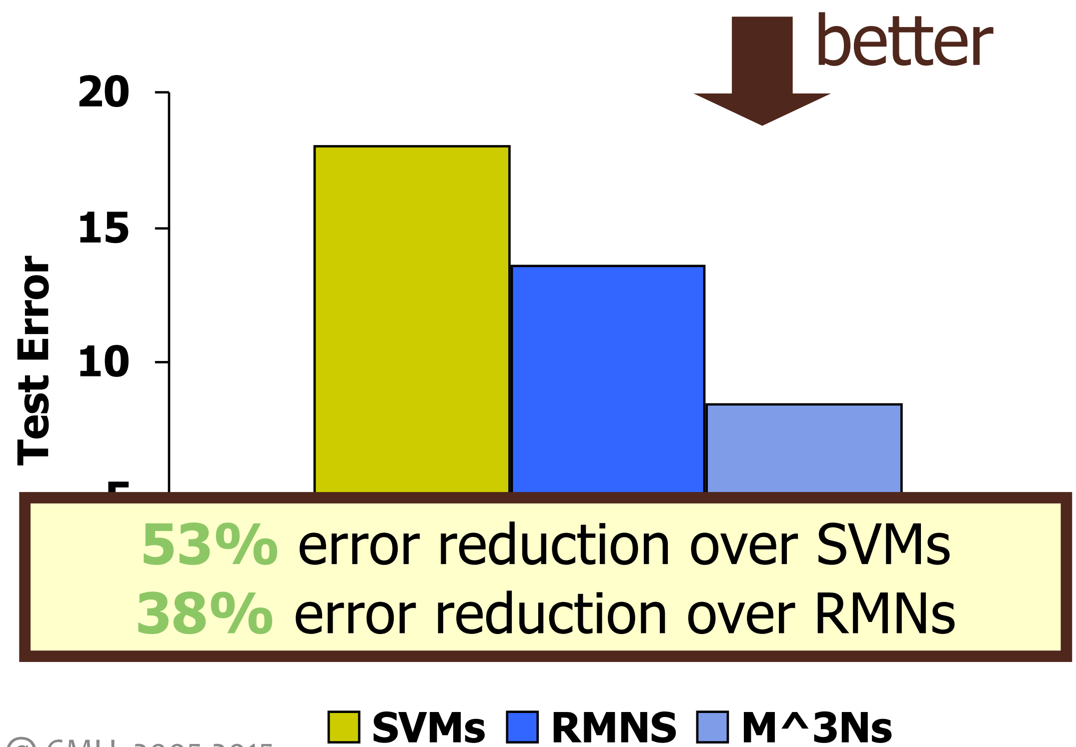
**33%** error reduction over multiclass SVMs

*Crammer & Singer 01

# Results: Hypertext Classification

- WebKB dataset
  - Four CS department websites: 1300 pages/3500 links
  - Classify each page: faculty, course, student, project, other
  - Train on three universities/test on fourth
- Inference: loopy belief propagation
- Learning: relaxed dual

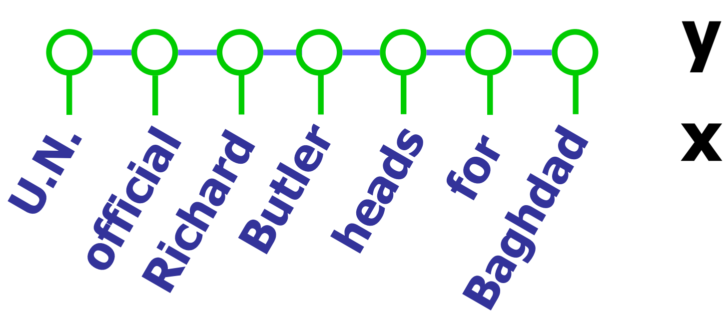





*Taskar et al 02

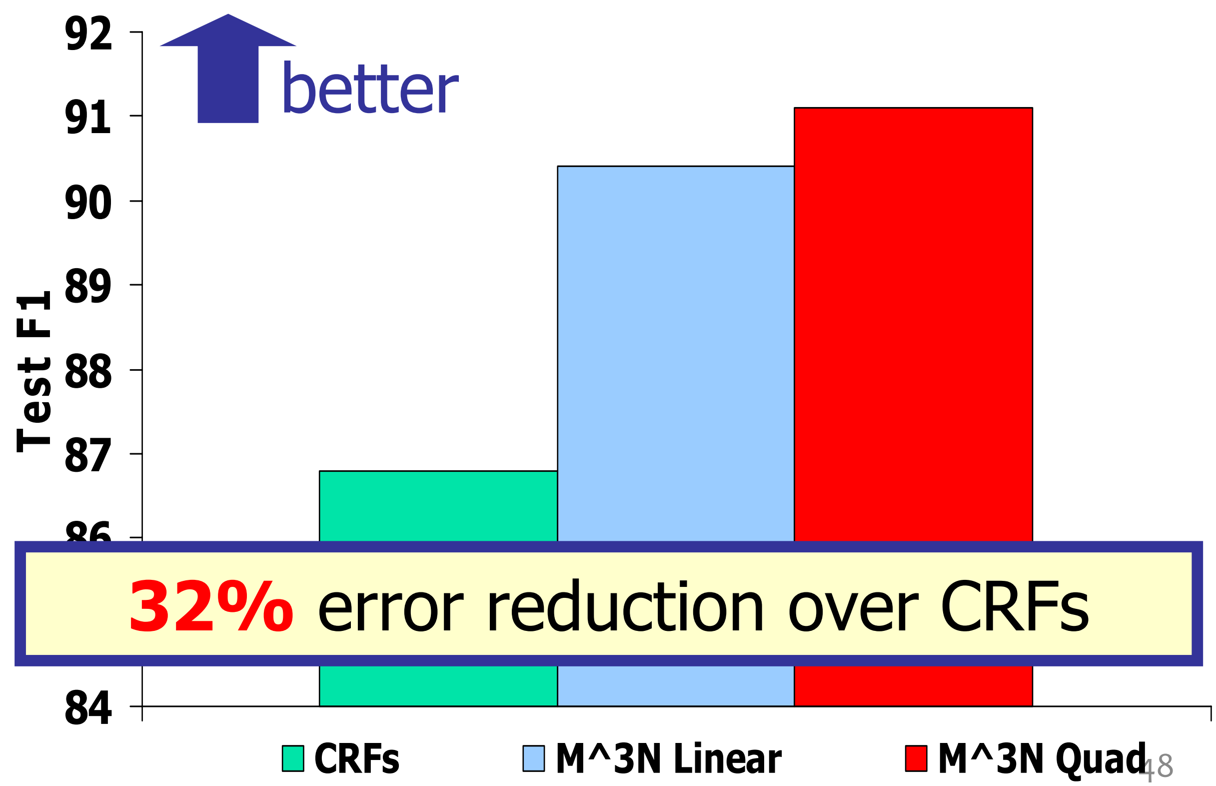
# Named Entity Recognition

- Locate and classify named entities in sentences:
  - 4 categories: organization, person, location, misc.
  - e.g. "U.N. official Richard Butler heads for Baghdad".
- CoNLL 03 data set (200K words train, 50K words test)




$y_i$ = org/per/loc/misc/none

$\mathbf{f}(y_i, x)$ = […,
I($y_i$=org, $x_i$="U.N."),
I($y_i$=per, $x_i$=capitalized),
I($y_i$=loc, $x_i$=known city),
…, ]




**32%** error reduction over CRFs