Max-Margin Markov Networks, B. Taskar, C. Guestrin and D. Koller. Neural Information Processing Systems Conference (NIPS03), 2003.
http://www.cs.berkeley.edu/~taskar/pubs/mmmn.ps
Learning Associative Markov Networks, B. Taskar, V. Chatalbashev and D. Koller. Twenty First International Conference on Machine Learning (ICML04), 2004.
http://www.cs.berkeley.edu/~taskar/pubs/mmamn.ps
Max-Margin Parsing, B. Taskar, D. Klein, M. Collins, D. Koller and C. Manning. Empirical Methods in Natural Language Processing (EMNLP04), 2004.
http://www.cs.berkeley.edu/~taskar/pubs/mmcfg.ps

Graphical Models meet Margin-based Learning

Machine Learning – 10701/15781

Carlos Guestrin

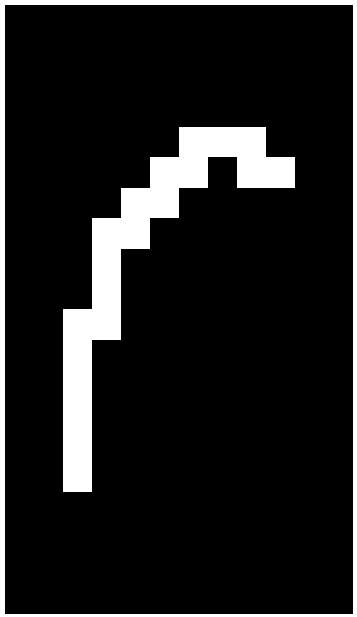
Carnegie Mellon University

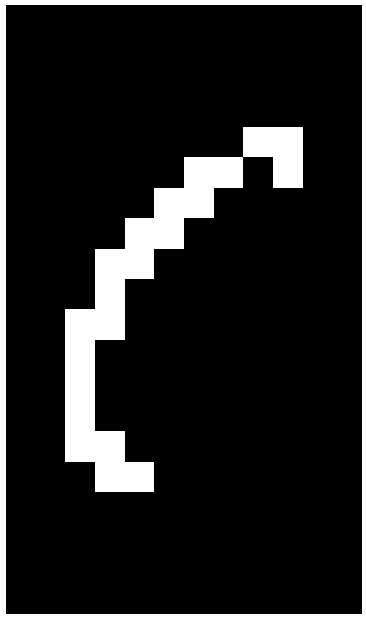
April 13th, 2005

Next few lectures

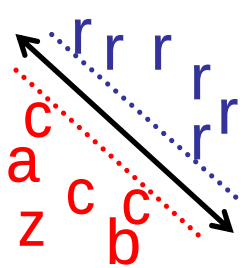
- Today – Advanced topic in graphical models
- Next week – learning to make decisions with reinforcement learning
- Week after – Dealing with very large datasets, active learning and BIG PICTURE

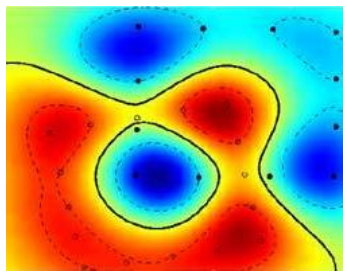
Handwriting Recognition





Character recognition: kernel SVMs

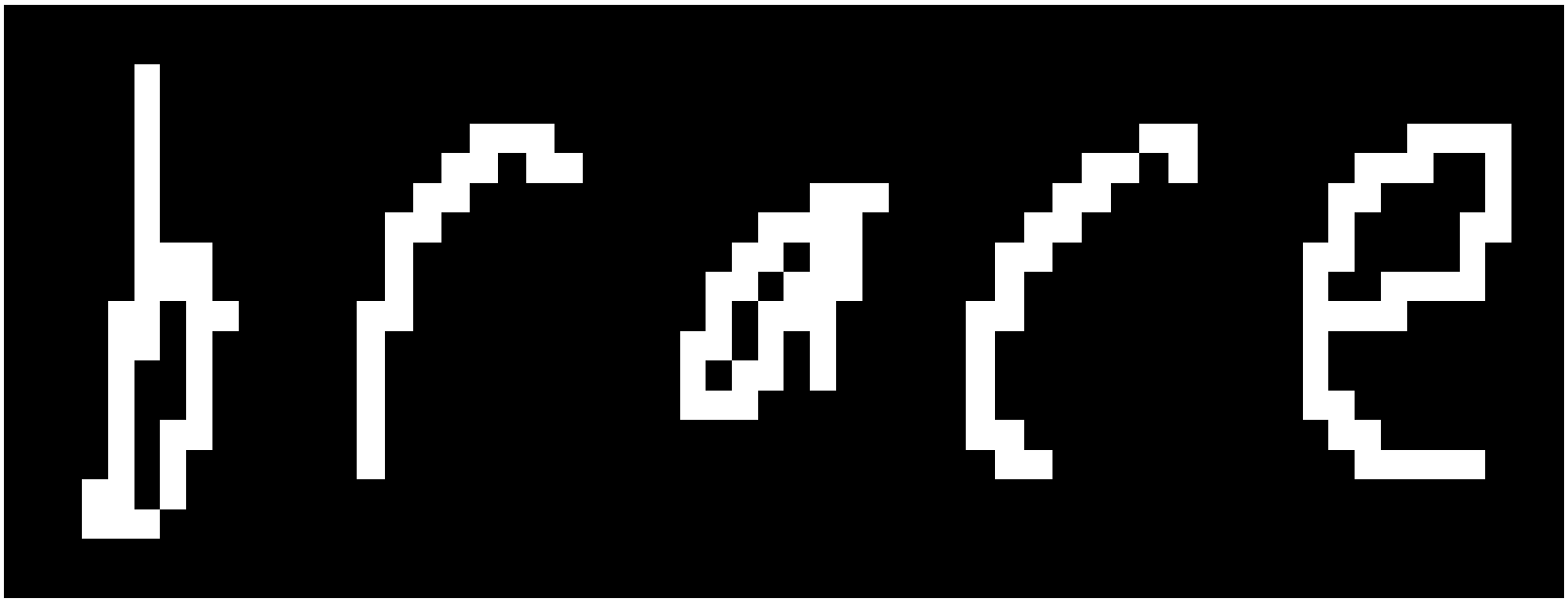




Support Vector Machines

Advantages:	SVM
High-dim learning (kernels)	
Generalization bounds	

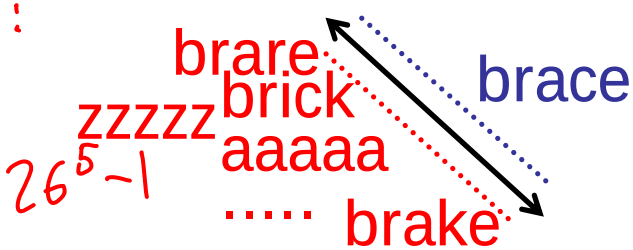
Handwriting Recognition 2



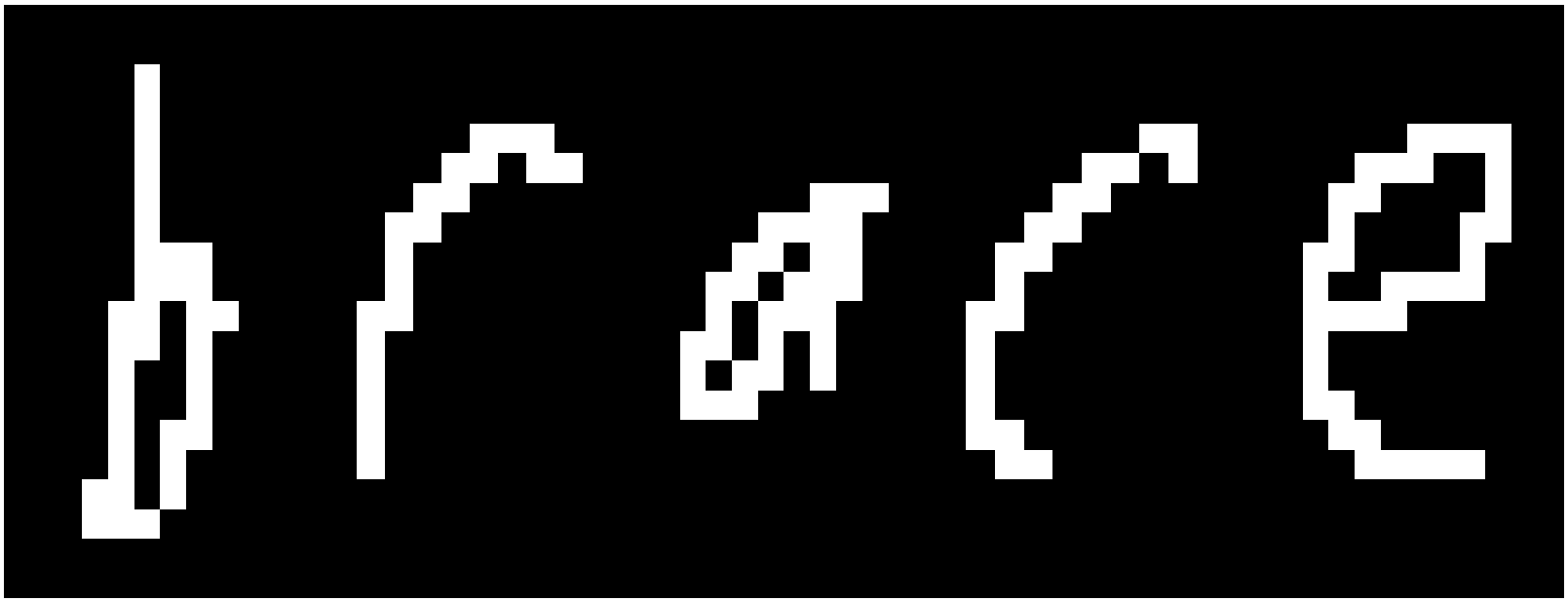
SVMs for sequences?

Problem: # of classes exponential in length

length word 5:

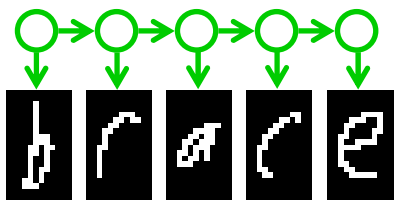


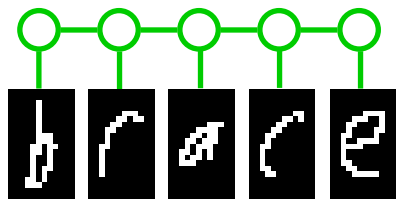
Handwriting Recognition 2



Graphical models: HMMs, MNs

HMM Linear in length Markov Network





SVMs vs. MNs

Advantages:	SVM	MN
High-dim learning (kernels)		
Generalization bounds		
Efficiently exploit label correlations		

SVMs, MNs vs. M^3Ns

Advantages:	SVM	MN	M^3N
High-dim learning (kernels)	✓	✗	✓
Generalization bounds	✓	✗	✓
Efficiently exploit label correlations	✗	✓	✓

conv of f: HMM: $P(y,x) = P(y_1)\left[\prod_{i=1} P(o_i|y_i)\right]\left[\prod_{i=1} P(y_{i+1}|y_i)\right]$ obs. prob. transition model

Chain Markov Net (aka CRF*)

word images

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

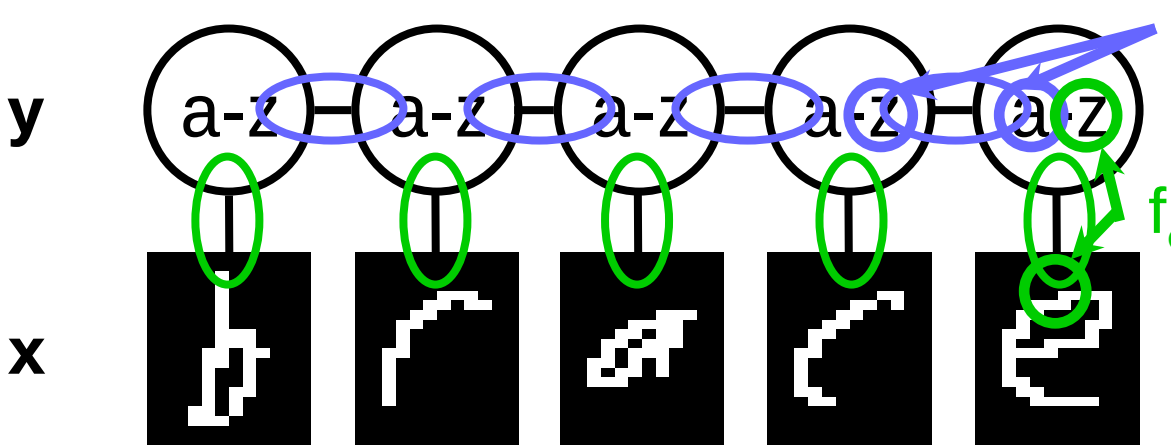
normalization constant

$\phi(\cdot) \geq 0$ potential function

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

linear function

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



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

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

*Lafferty et al. 01

$e^{\sum_\alpha w_\alpha f_\alpha} \cdot e^{\sum_\beta w_\beta f_\beta} = e^{\sum_\alpha w_\alpha f_\alpha + \sum_\beta w_\beta f_\beta}$

Chain Markov Net (aka CRF*)

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

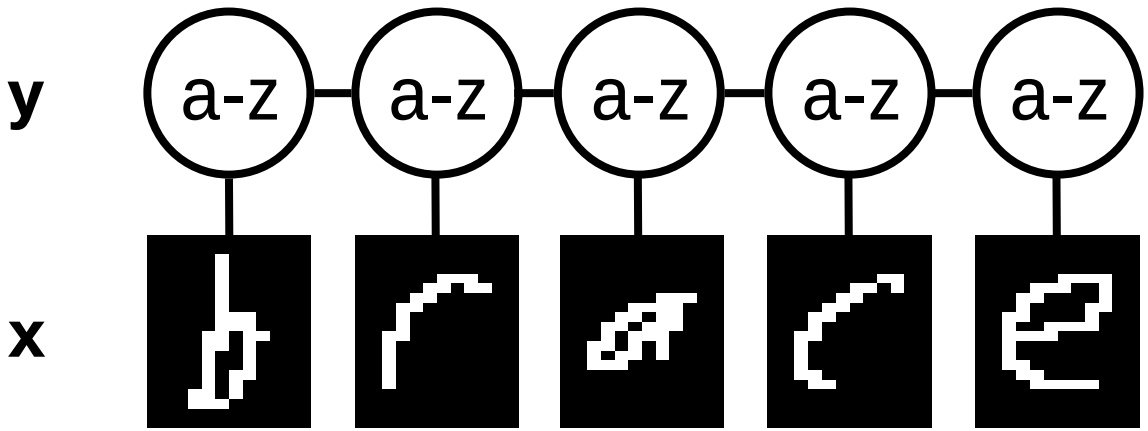
log-linear model

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

$$\mathbf{w} = [\ldots, w_\alpha, \ldots, w_\beta, \ldots]$$

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

$$\mathbf{f}(\mathbf{x},\mathbf{y}) = [\ldots, f_\alpha(\mathbf{x},\mathbf{y}), \ldots, f_\beta(\mathbf{x},\mathbf{y}), \ldots]$$



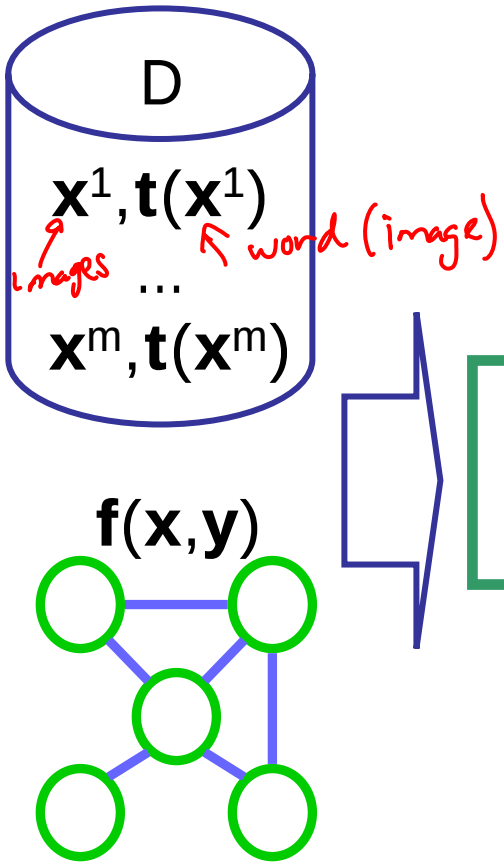
$f_\beta(\mathbf{x},\mathbf{y}) = \#(y='z', y'='a')$

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

*Lafferty et al. 01

Max (Conditional) Likelihood

$$P_w(y|x) = \frac{1}{Z(x)} e^{w^T f(x,y)}$$



Estimation

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

Classification

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

word, image

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

doesn't matter in argmax

Don't need to learn entire distribution!

OCR Example

- We want:

  $\text{argmax}_{\textbf{word}}$ $\mathbf{w}^T \mathbf{f}($ $,\textbf{word}) = \text{“brace”}$

- Equivalently:

  $\mathbf{w}^T \mathbf{f}($ $,\text{“brace”}) > \mathbf{w}^T \mathbf{f}($ $,\text{“aaaaa”})$

  $\mathbf{w}^T \mathbf{f}($ $,\text{“brace”}) > \mathbf{w}^T \mathbf{f}($ $,\text{“aaaab”})$

  …

  $\mathbf{w}^T \mathbf{f}($ $,\text{“brace”}) > \mathbf{w}^T \mathbf{f}($ $,\text{“zzzzz”})$

  a lot!

  $26^5 - 1$

Max Margin Estimation

- Goal: find **w** such that

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{t}(\mathbf{x})) > \mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{y}) \qquad \forall \mathbf{x} \in D \;\; \forall \mathbf{y} \neq \mathbf{t}(\mathbf{x})$$

$$\mathbf{w}^\top [\mathbf{f}(\mathbf{x},\mathbf{t}(\mathbf{x})) - \mathbf{f}(\mathbf{x},\mathbf{y})] > 0$$

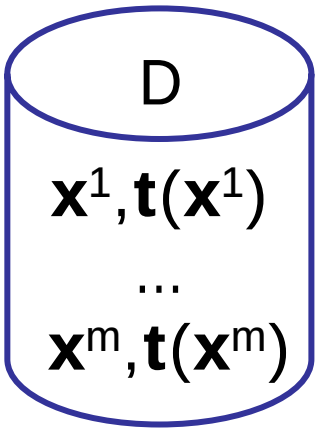
$$\mathbf{w}^\top \Delta\mathbf{f}_{\mathbf{x}}(\mathbf{y}) \geq \gamma \Delta\mathbf{t}_{\mathbf{x}}(\mathbf{y})$$

- Maximize margin γ
- Gain over **y** grows with # of mistakes in **y**: $\Delta\mathbf{t}_{\mathbf{x}}(\mathbf{y})$

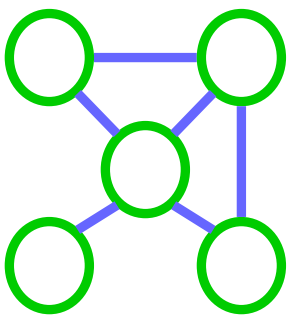
$\Delta\mathbf{t}_{\text{brace}}(\text{“craze”}) = 2$ $\qquad$ $\Delta\mathbf{t}_{\text{brace}}(\text{“zzzzz”}) = 5$

$\mathbf{w}^\top \Delta\mathbf{f}_{\text{brace}}(\text{“craze”}) \geq 2\gamma$ $\qquad$ $\mathbf{w}^\top \Delta\mathbf{f}_{\text{brace}}(\text{“zzzzz”}) \geq 5\gamma$

M^{3}Ns



f(x,y)



Estimation

$$\max_{||w|| \leq 1} \quad \gamma$$
$$\mathbf{w}^\top \Delta \mathbf{f}_\mathbf{x}(\mathbf{y}) \geq \gamma \Delta \mathbf{t}_\mathbf{x}(\mathbf{y})$$

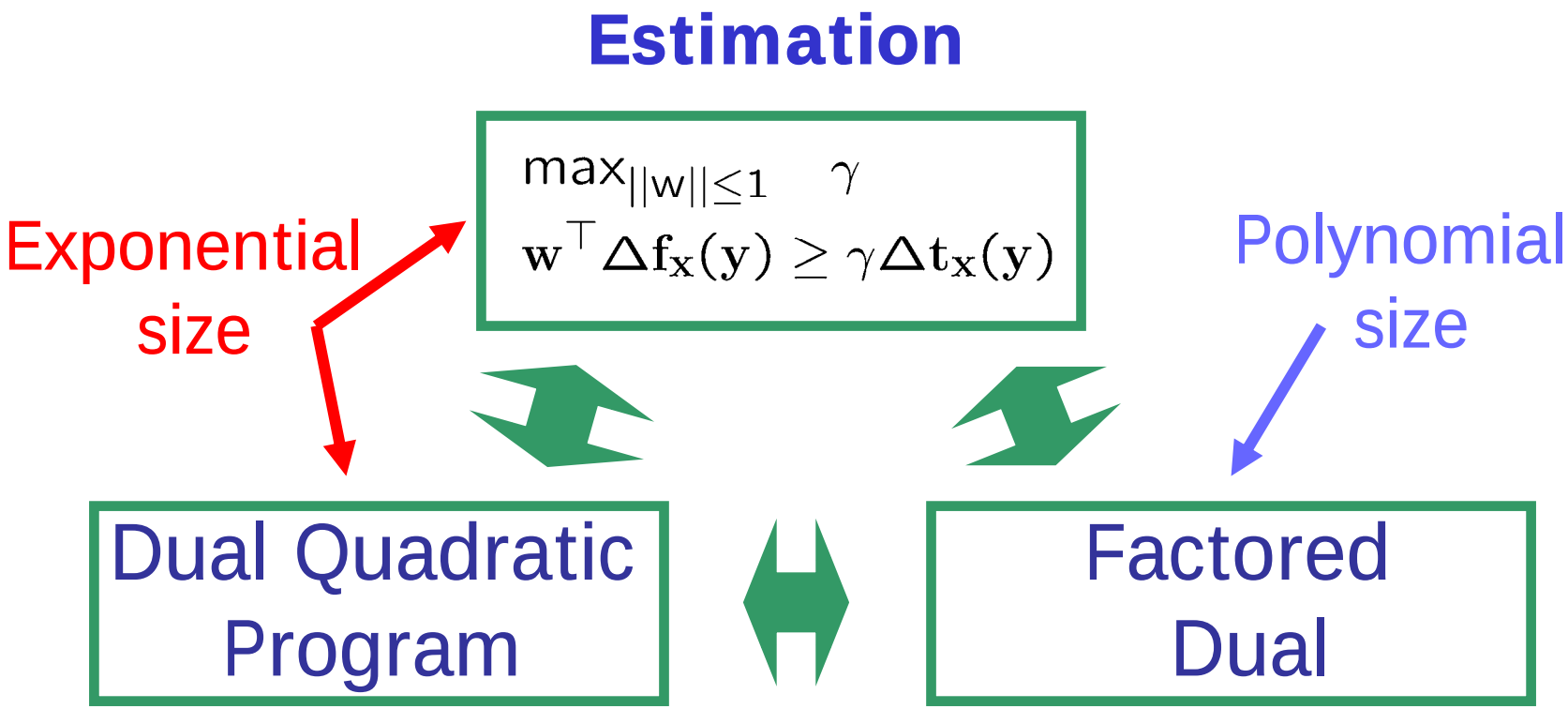
Classification

new image

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

wordy

M^{3}Ns


Estimation
$\max_{||w|| \leq 1} \quad \gamma$
$\mathbf{w}^\top \Delta \mathbf{f_x(y)} \geq \gamma \Delta \mathbf{t_x(y)}$
Exponential size
Polynomial size
Dual Quadratic Program
Factored Dual

M³N Dual

$\mathbf{w}^\top \Delta \mathbf{f}_{brace}(\text{“craze”}) \geq 2\gamma \rightarrow \alpha_{brace}(\text{“craze”})$

$\mathbf{w}^\top \Delta \mathbf{f}_{brace}(\text{“zzzzz”}) \geq 5\gamma \rightarrow \alpha_{brace}(\text{“zzzzz”})$

$$\max_{\alpha} \quad \sum_{\mathbf{x}} \sum_{\mathbf{y}} \alpha_{\mathbf{x}}(\mathbf{y}) \Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y}) - \frac{1}{2} \sum_{\mathbf{x},\mathbf{x}'} \sum_{\mathbf{y},\mathbf{y}'} \alpha_{\mathbf{x}}(\mathbf{y}) \alpha_{\mathbf{x}'}(\mathbf{y}') \Delta \mathbf{f}_{\mathbf{x}}(\mathbf{y})^\top \Delta \mathbf{f}_{\mathbf{x}'}(\mathbf{y}')$$

$$\text{s.t.} \quad \sum_{\mathbf{y}} \alpha_{\mathbf{x}}(\mathbf{y}) = 1 \text{ and } \alpha_{\mathbf{x}}(\mathbf{y}) \geq 0 \quad \forall \mathbf{x} \in D \; \forall \mathbf{y}$$

26^5 $\alpha_x(y)$ $|y| = 26^5$

- Exponential number of variables
  - $\alpha_{\mathbf{x}}(\mathbf{y})$ represents a probability distribution
- Key insight from graphical models:
  - Can use network structure to factorize distribution

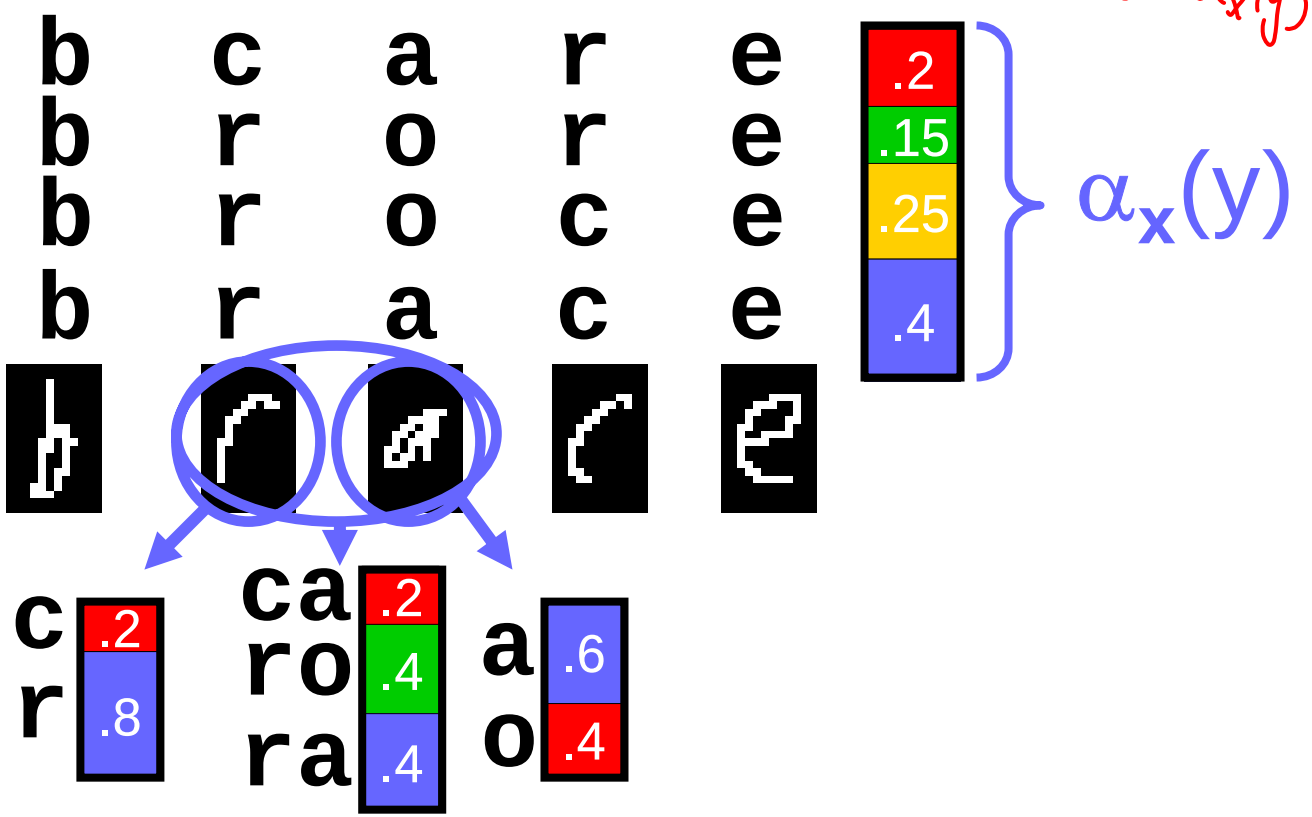
Dual = Probability Distribution

features

$$\max_{\alpha} \quad \sum_{x}\sum_{y} \alpha_x(y)\Delta t_x(y) - \frac{1}{2}\sum_{x,x'}\sum_{y,y'} \alpha_x(y)\alpha_{x'}(y')\Delta f_x(y)^\top \Delta f_{x'}(y')$$

$$\text{s.t.} \quad \sum_{y} \alpha_x(y) = 1 \;\text{ and }\; \alpha_x(y) \geq 0 \quad \forall x \in D \; \forall y$$

26 $\alpha_x(y)$



Factored Dual Variables

- Introduce factored dual variables:

$$\mu_i(y_i) \equiv \sum_{\mathbf{y} \sim y_i} \alpha(\mathbf{y}) \qquad \mu_{ij}(y_i, y_j) \equiv \sum_{\mathbf{y} \sim y_i, y_j} \alpha(\mathbf{y})$$

  - Linear in the size of the network

- Rewrite dual using μ's:

  maximize QuadraticObjective(μ)

  s.t. $\mu \in$ ConsistentMarginals (linear constraints)

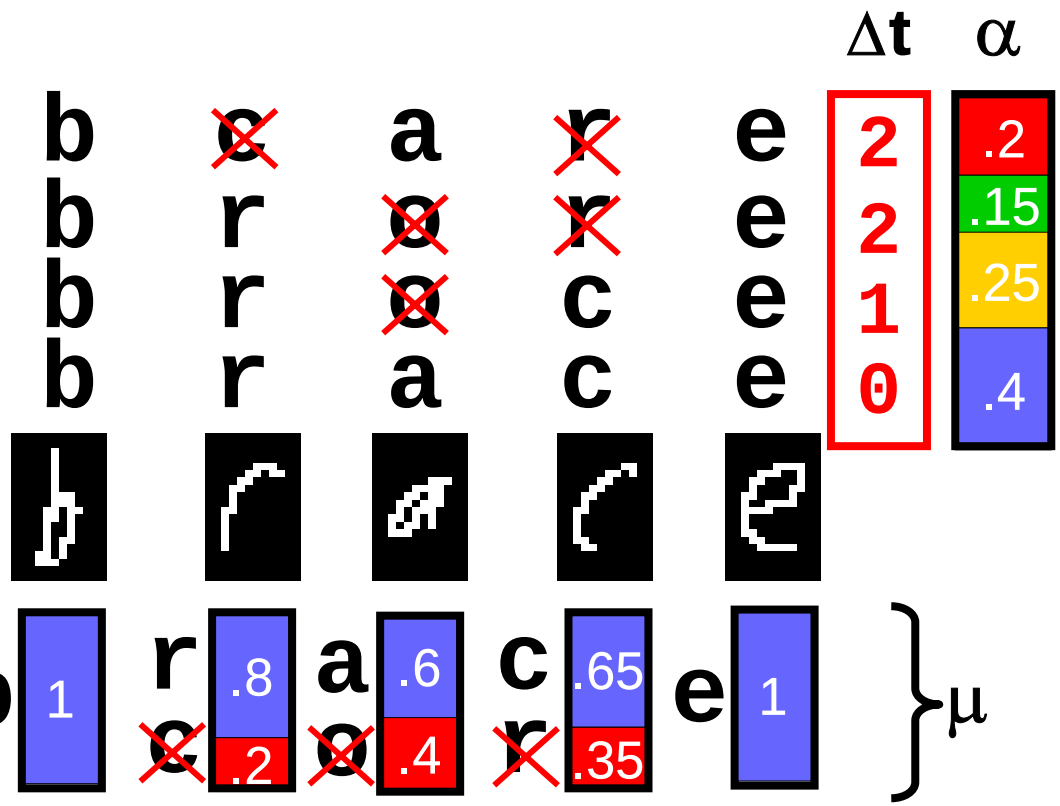
Equivalent to original dual!

Factored Objective

$$\underbrace{\sum_{\mathbf{y}} \alpha(\mathbf{y})\Delta\mathbf{t}(\mathbf{y})}_{E_\alpha[\Delta\mathbf{t}(\mathbf{y})]} - \frac{1}{2}\sum_{\mathbf{y},\mathbf{y}'} \alpha(\mathbf{y})\alpha(\mathbf{y}')\Delta\mathbf{f}(\mathbf{y})^\top \Delta\mathbf{f}(\mathbf{y}')$$

$$\Delta\mathbf{t}(\mathbf{y}) = \sum_i \Delta\mathbf{t}(y_i)$$

$$E_\alpha[\Delta\mathbf{t}(\mathbf{y})] = E_\mu[\Delta\mathbf{t}(\mathbf{y})]$$



Factored Objective

$$\underbrace{\sum_{\mathbf{y}} \alpha(\mathbf{y}) \Delta \mathbf{t}(\mathbf{y})}_{} - \frac{1}{2} \underbrace{\sum_{\mathbf{y},\mathbf{y}'} \alpha(\mathbf{y}) \alpha(\mathbf{y}') \Delta \mathbf{f}(\mathbf{y})^\top \Delta \mathbf{f}(\mathbf{y}')}_{}$$

$$E_\alpha[\Delta \mathbf{t}(\mathbf{y})] - \frac{1}{2} E_\alpha[\Delta \mathbf{f}(\mathbf{y})]^\top E_\alpha[\Delta \mathbf{f}(\mathbf{y})]$$

$$\Delta \mathbf{t}(\mathbf{y}) = \sum_i \Delta \mathbf{t}(y_i)$$

$$\Delta \mathbf{f}(\mathbf{y}) = \sum_i \Delta \mathbf{f}(y_i) + \sum_{ij} \Delta \mathbf{f}(y_i, y_j)$$

$$E_\alpha[\Delta \mathbf{t}(\mathbf{y})] = E_\mu[\Delta \mathbf{t}(\mathbf{y})]$$

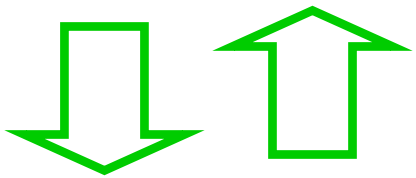
$$E_\alpha[\Delta \mathbf{f}(\mathbf{y})] = E_\mu[\Delta \mathbf{f}(\mathbf{y})]$$

$$E_\mu[\Delta \mathbf{t}(\mathbf{y})] - \frac{1}{2} E_\mu[\Delta \mathbf{f}(\mathbf{y})]^\top E_\mu[\Delta \mathbf{f}(\mathbf{y})]$$

Factored Constraints

$$\sum_{\mathbf{y}} \alpha(\mathbf{y}) = 1$$ normalization

$$\alpha(\mathbf{y}) \geq 0$$ non-negativity



If network is a tree

Else add clique tree constraints

$$\sum_{y_i} \mu(y_i) = 1 \qquad \sum_{y_i, y_j} \mu(y_i, y_j) = 1$$ normalization

$$\mu(y_i) \geq 0 \qquad \mu(y_i, y_j) \geq 0$$ non-negativity

$$\mu(y_i) = \sum_{y_j} \mu(y_i, y_j)$$ agreement

$$\mu(\cdot) \in \text{CliqueTreePolytope}$$ triangulation

Factored Dual

$$\max_{\mu} \quad \sum_{\mathbf{x}} E_{\mu_{\mathbf{x}}}[\Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y})] - \frac{1}{2} \sum_{\mathbf{x},\mathbf{x}'} E_{\mu_{\mathbf{x}}}[\Delta \mathbf{f}_{\mathbf{x}}(\mathbf{y})]^{\top} E_{\mu_{\mathbf{x}'}}[\Delta \mathbf{f}_{\mathbf{x}'}(\mathbf{y}')]$$

$$\text{s.t.} \quad \sum_{\mathbf{y}_i} \mu_{\mathbf{x}}(\mathbf{y}_i) = 1 \qquad \mu_{\mathbf{x}}(\mathbf{y}_i, \mathbf{y}_j) \geq 0 \qquad \mu_{\mathbf{x}}(\mathbf{y}_i) = \sum_{\mathbf{y}_i} \mu_{\mathbf{x}}(\mathbf{y}_i, \mathbf{y}_j)$$

$$\mu_{\mathbf{x}}(\cdot) \in \text{CliqueTreePolytope}_{\mathbf{x}}$$

- Objective is quadratic in network size
- Constraint set is exponential in tree-width
  - Linear for sequences and trees
  - Complexity same as inference and max likelihood

Factored Dual

$$\max_\mu \quad \sum_{\mathbf{x}} E_{\mu_{\mathbf{x}}}[\Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y})] - \frac{1}{2} \sum_{\mathbf{x},\mathbf{x}'} E_{\mu_{\mathbf{x}}}[\Delta \mathbf{f}_{\mathbf{x}}(\mathbf{y})]^\top E_{\mu_{\mathbf{x}'}}[\Delta \mathbf{f}_{\mathbf{x}'}(\mathbf{y}')]$$

$$\text{nodes} \Rightarrow \sum_{i,y_i} \sum_{k,y'_k} \mu_{\mathbf{x}}(y_i)\mu_{\mathbf{x}'}(y'_k)\Delta \mathbf{f}_{\mathbf{x}}(y_i)^\top \Delta \mathbf{f}_{\mathbf{x}'}(y'_k) +$$

$$\text{edges} \Rightarrow \sum_{\substack{ij \\ y_i,y_j}} \sum_{\substack{km \\ y'_k,y'_m}} \mu_{\mathbf{x}}(y_i,y_j)\mu_{\mathbf{x}'}(y'_k,y'_m)\Delta \mathbf{f}_{\mathbf{x}}(y_i,y_j)^\top \Delta \mathbf{f}_{\mathbf{x}'}(y'_k,y'_m)$$

$\phi(x_i, y_i)$

$\phi(y_i, y_{i+1})$

- Kernel trick works!
  - Node and edge potentials can use kernels

Generalization Bound

Theorem:

Per-label loss $\mathcal{L}$ for m training examples:

$$E_{\mathbf{x}}\mathcal{L}(\mathbf{w},\mathbf{x}) \leq E_S\mathcal{L}^{\gamma}(\mathbf{w},\mathbf{x}) + \sqrt{\frac{K}{m}\left[\frac{||\Delta\mathbf{f}||^2||\mathbf{w}||^2}{\gamma^2}[\ln m + \ln L] + \ln\frac{1}{\delta}\right]}$$

Test set per-label error

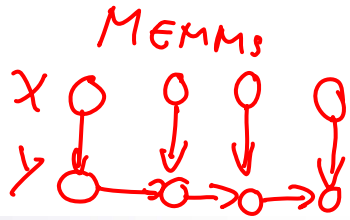
Training set per-label γ-error

better as margin increases

with probability at least 1-δ.

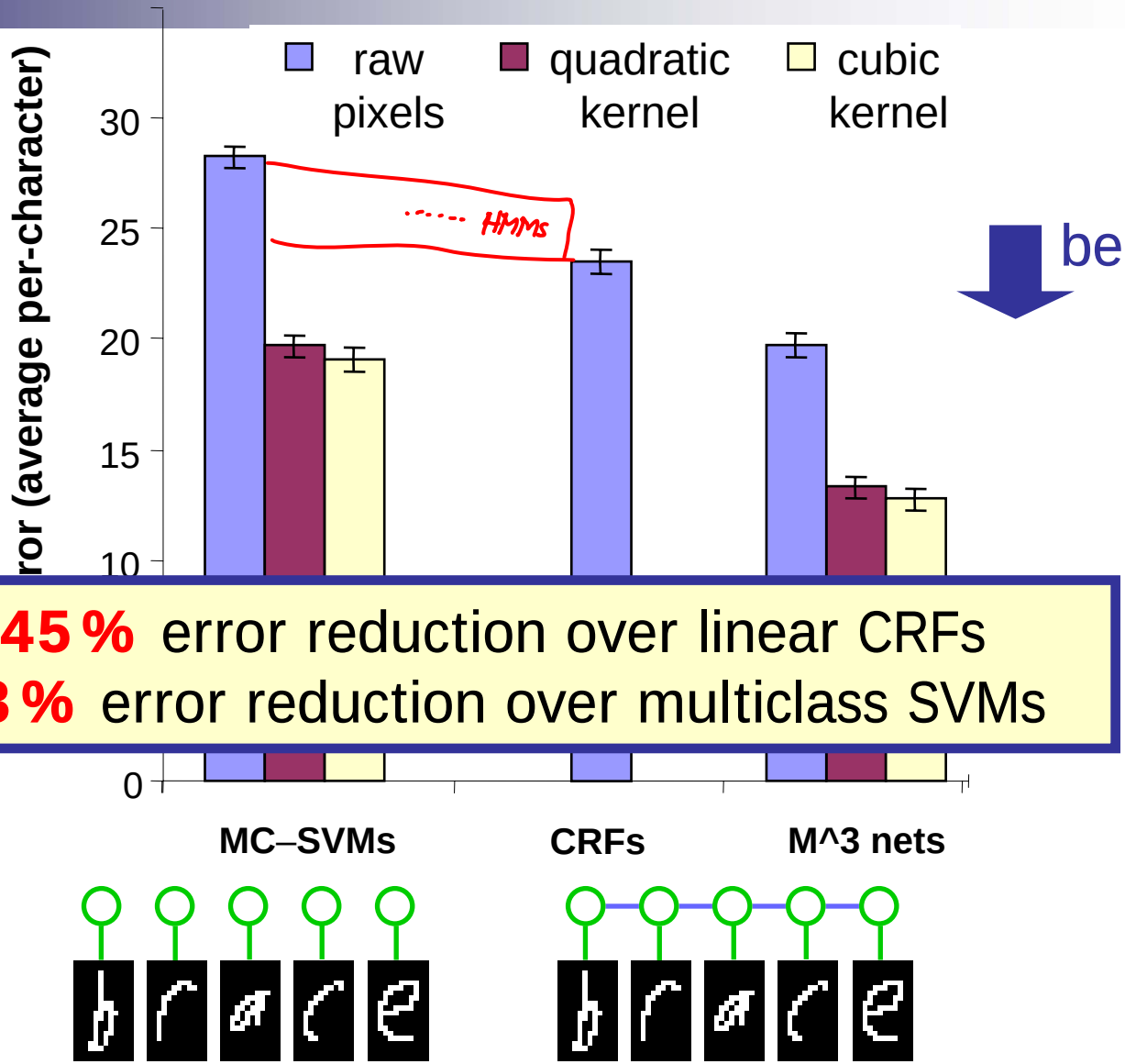
- Distribution-free
- First per-label bound
- Dependence on L logarithmic vs. linear [Colllins 01]
  - ***L* = number of nodes and edges**

Handwriting Recognition



Length: ~8 chars
Letter: 16x8 pixels
10-fold Train/Test
5000/50000 letters
600/6000 words

Models:
- Multiclass-SVM
- CRFs
- M^3 nets

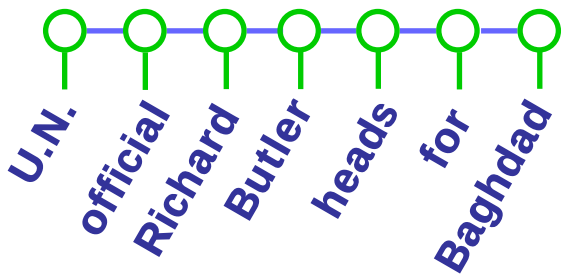


45% error reduction over linear CRFs
33% error reduction over multiclass SVMs

*Crammer & Singer 01

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)

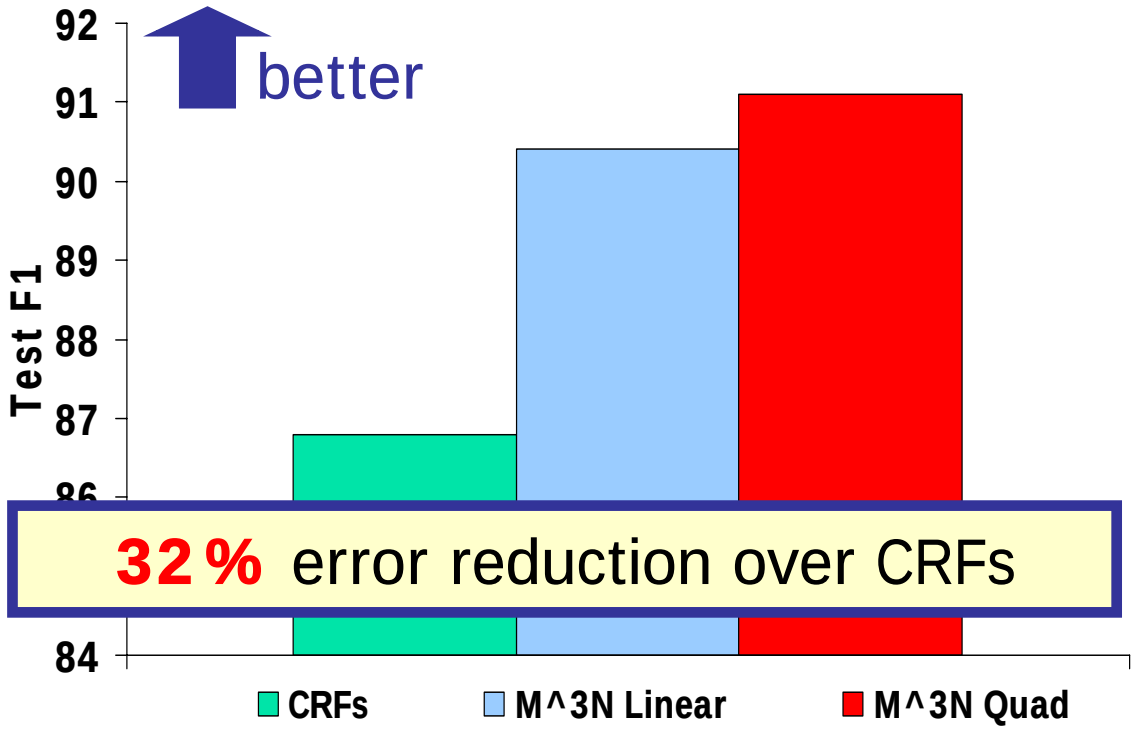


y

x

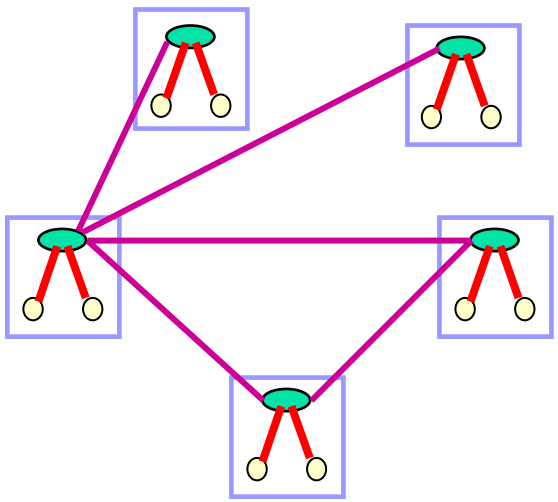
y_i = org/per/loc/misc/none

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),
...,]

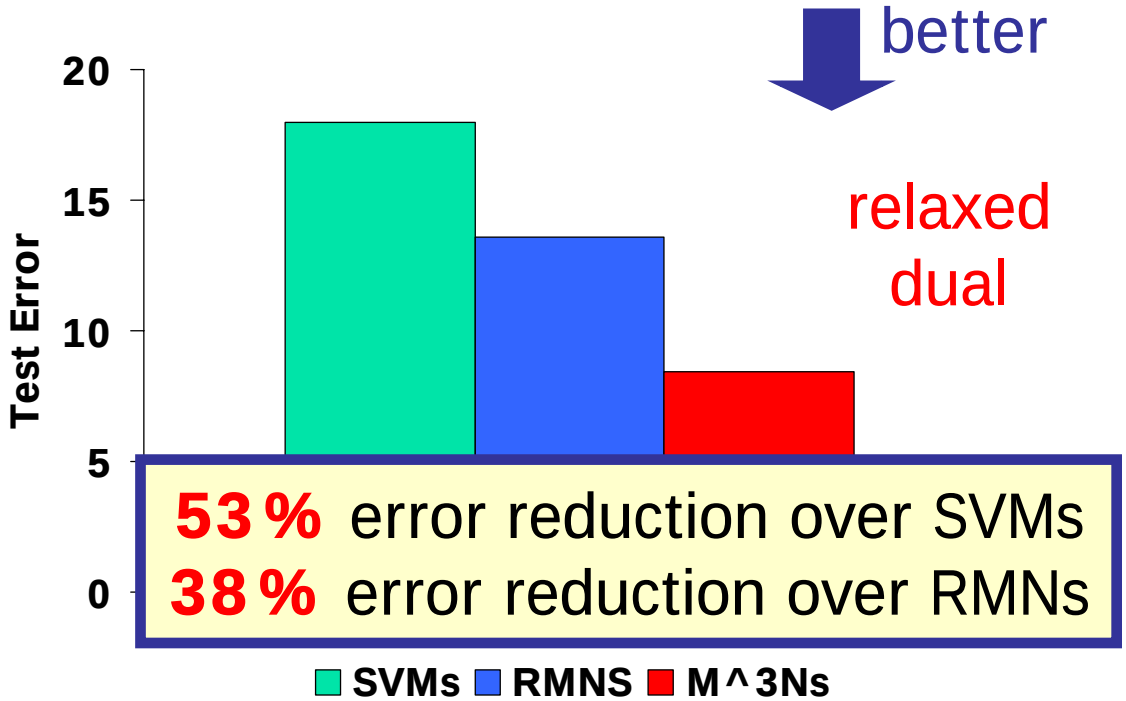


Hypertext Classification

- 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



loopy belief propagation



*Taskar et al 02

M^{3}Ns

Basic algorithm works for any low tree-width graphical model

Estimation

$$\max_{||w|| \leq 1} \quad \gamma$$
$$\mathbf{w}^\top \Delta \mathbf{f_x(y)} \geq \gamma \Delta \mathbf{t_x(y)}$$

Exponential size

Polynomial size

Dual Quadratic Program

Factored Dual

Other possible max-margin learning problems

- Large tree-width Markov networks with attractive potentials
- Parsing using probabilistic context-free grammars
- Learning to cluster
- Max-margin learning of any poly-time problem..

Associative Markov networks

$$P(\mathbf{y} \mid \mathbf{x}) \propto \underbrace{\prod_i \phi_i(y_i, \mathbf{x}_i)}_{\text{Point features}} \underbrace{\prod_{ij} \phi_{ij}(y_i, y_j, \mathbf{x}_{ij})}_{\text{Edge features}} = \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})\}$$

spin-images, point height

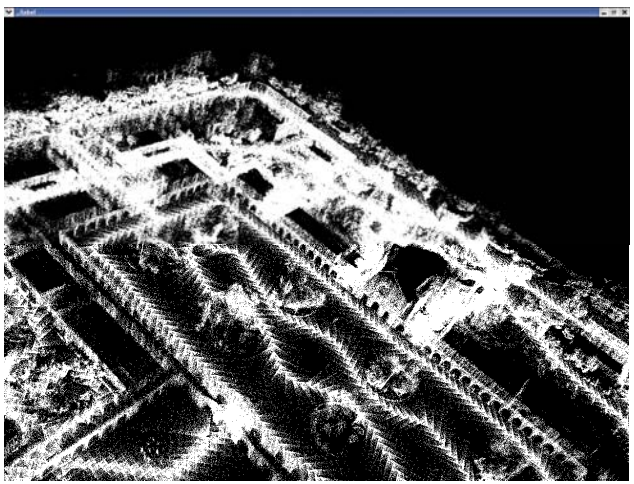
length of edge, edge orientation

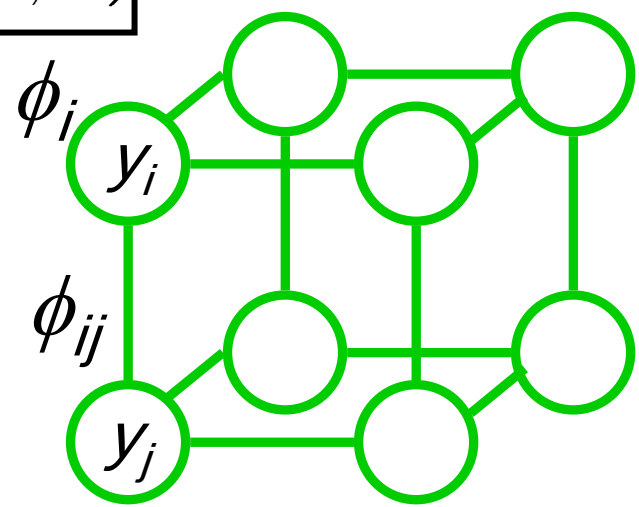
"associative" restriction

$$\phi_{ij}(y_i, y_j) = \begin{bmatrix} \phi_{ij}(1,1) & & 1 \\ & \ddots & \\ 1 & & \phi_{ij}(K,K) \end{bmatrix}$$

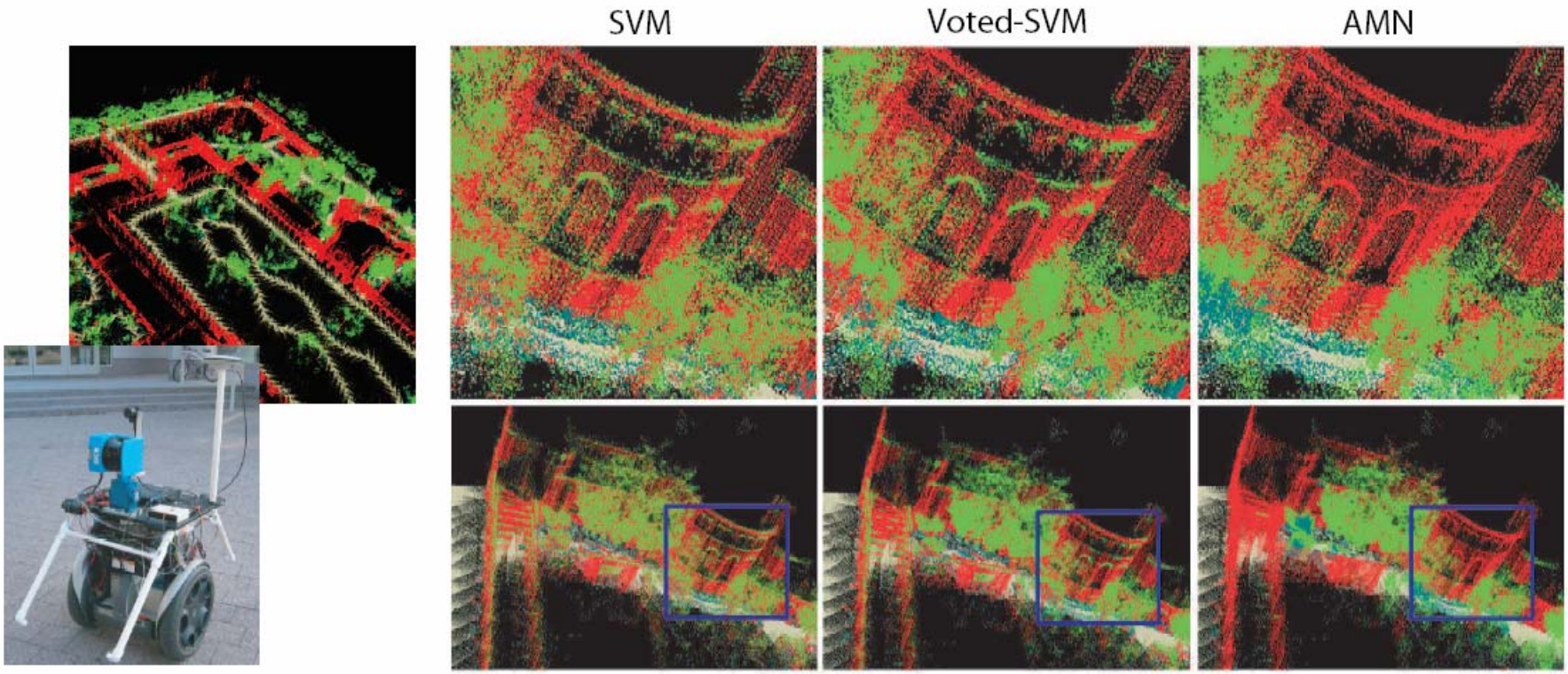
bonus

$$\phi_{ij}(k,k) \geq 1$$



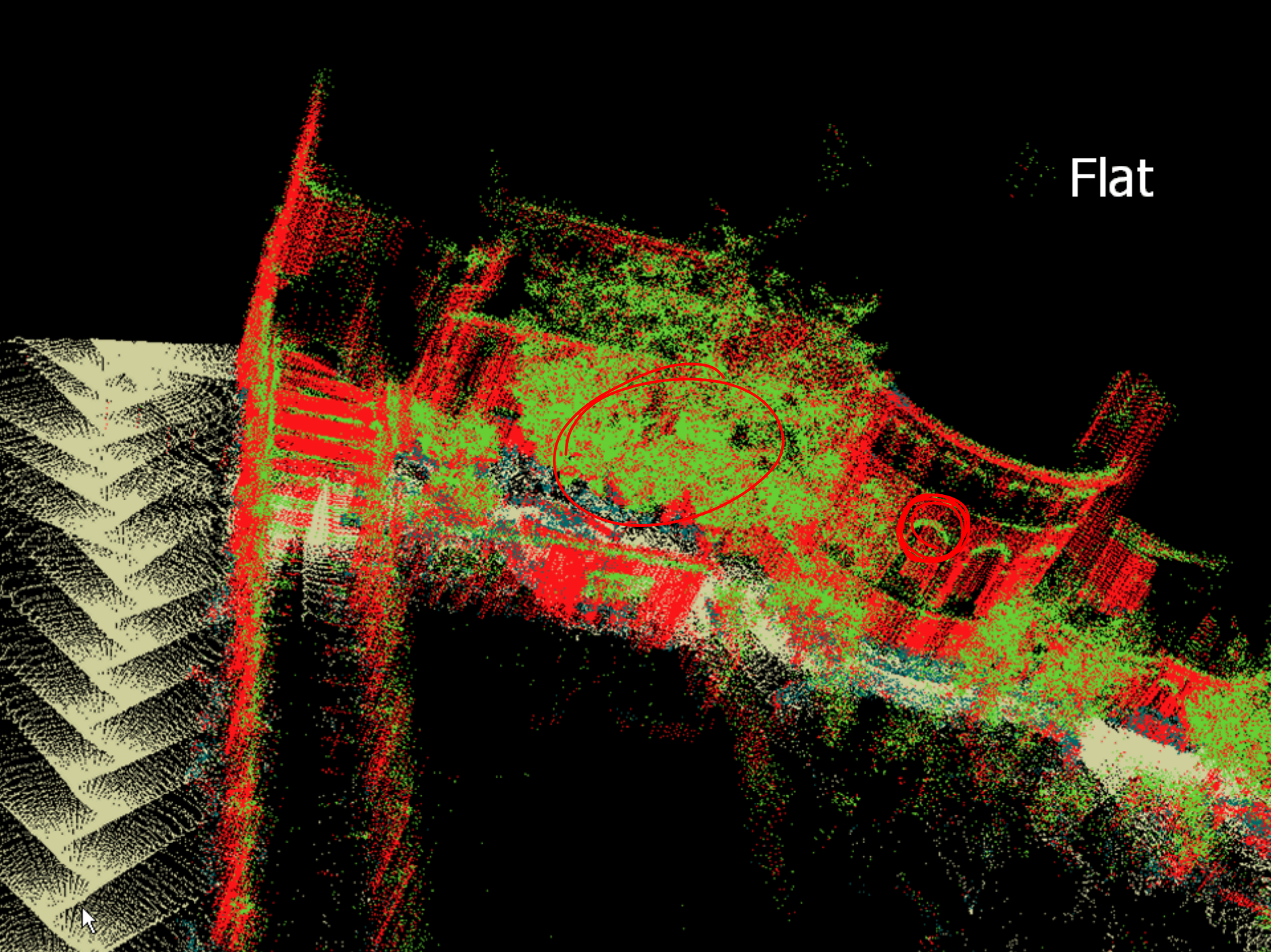


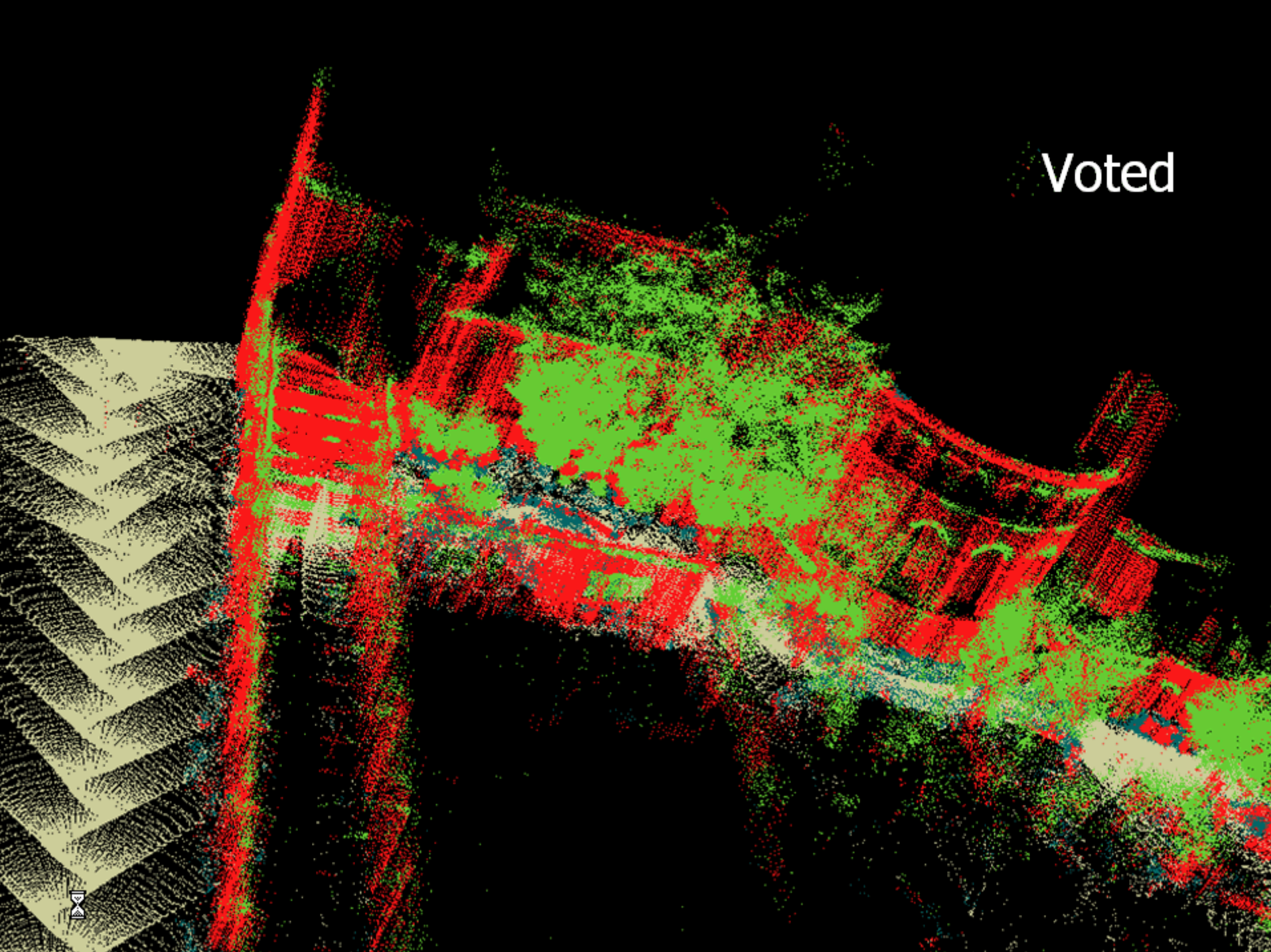
Max-margin AMNs results

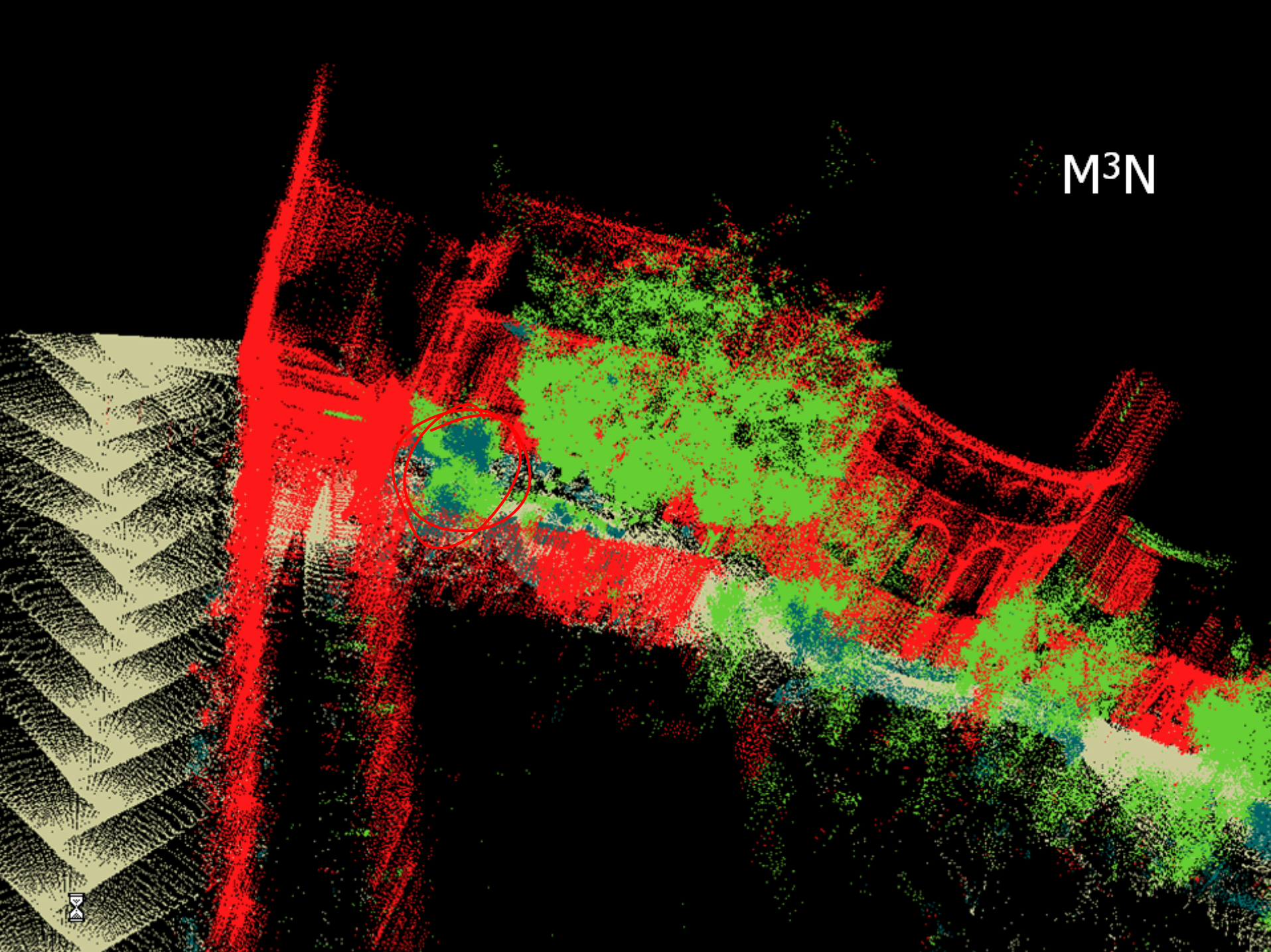


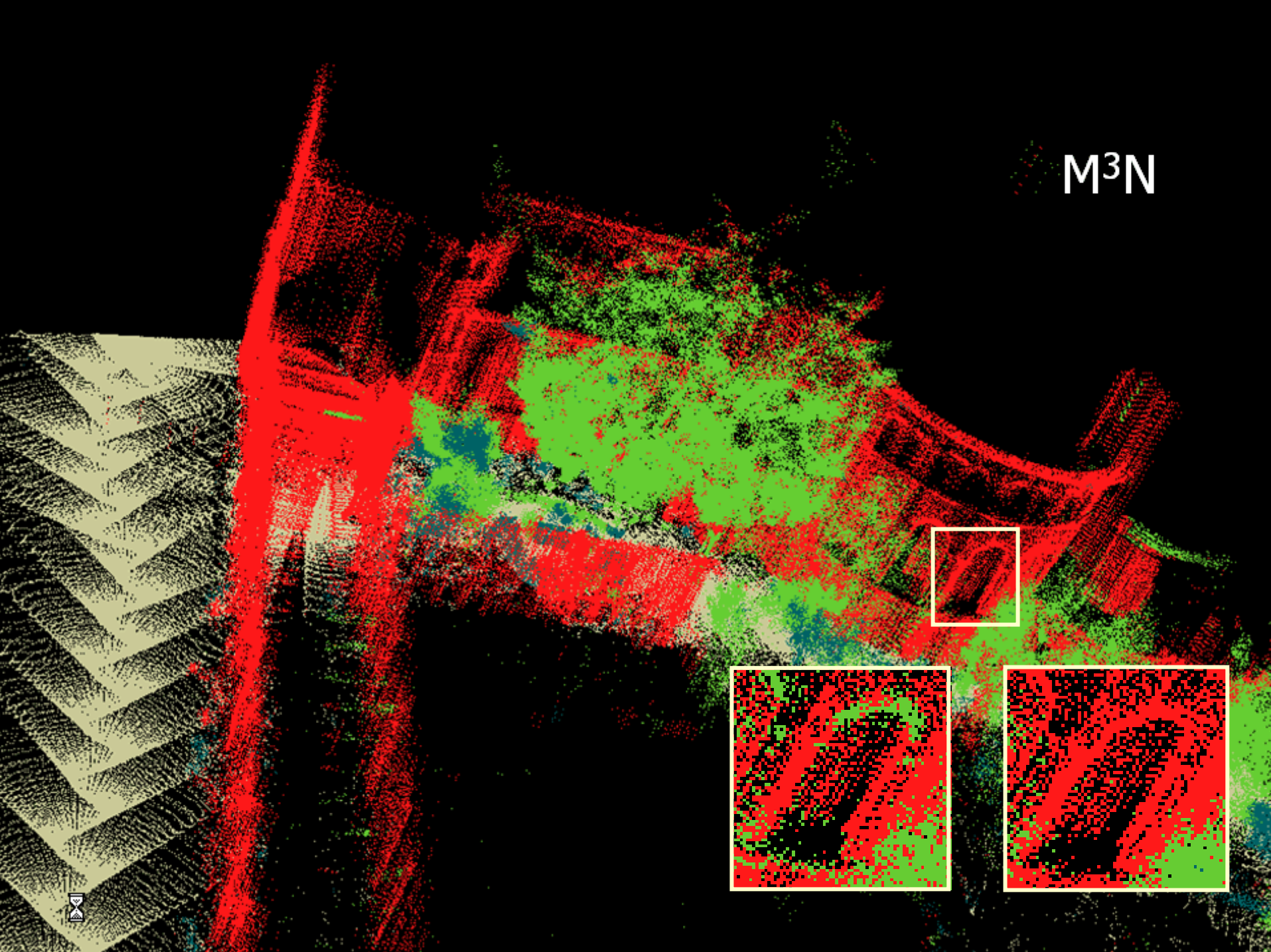
Label: ground, building, tree, shrub

Training: 30 thousand points Testing: 3 million points


Flat


Voted

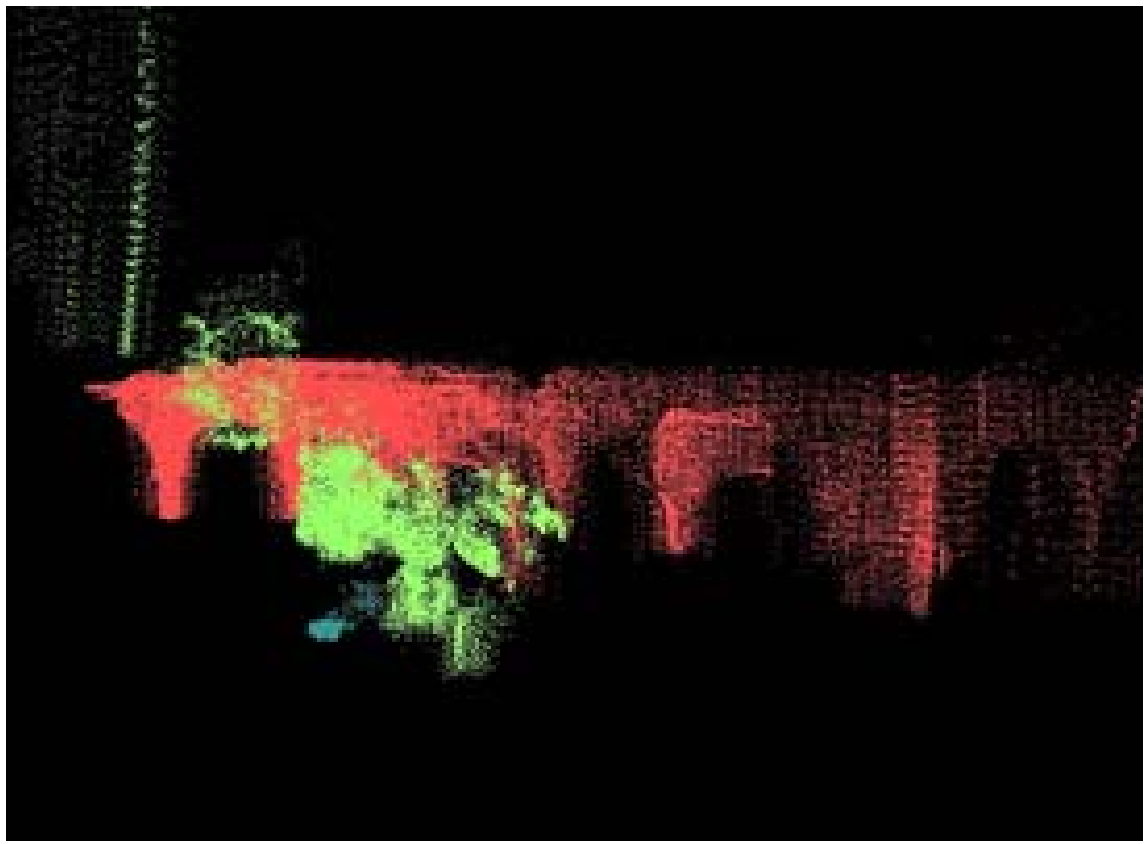

M3N


M3N

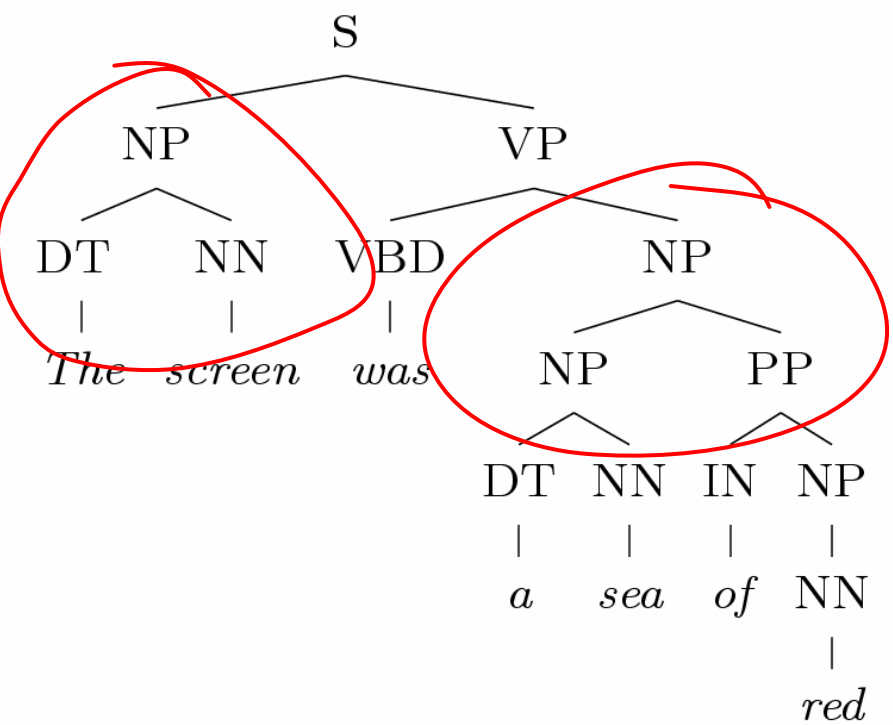
Segmentation results

Hand labeled 180K test points

Model	Accuracy
SVM	68%
V-SVM	73%
M^3N	93%



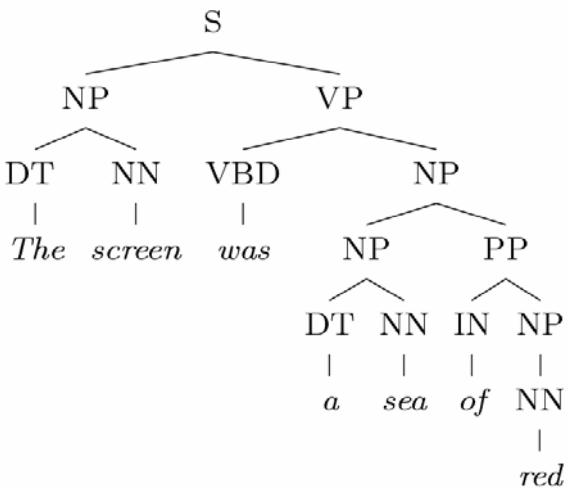
Max-margin parsing



- Classic learning problem:
  - P(NP→{NP,PP})
  - P(NP→{DT,NN})
  - ...
- Usually, learn probabilities with counts
- Learn max-margin discriminative model

PCFG

$$P(\mathbf{y} \mid \mathbf{x}) \propto \prod_{A \to \alpha \in (\mathbf{x},\mathbf{y})} P(A \to \alpha) = \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})\}$$



$$\mathbf{f} : \mathcal{X} \times \mathcal{Y} \to \Re^d$$

#(NP → DT NN)

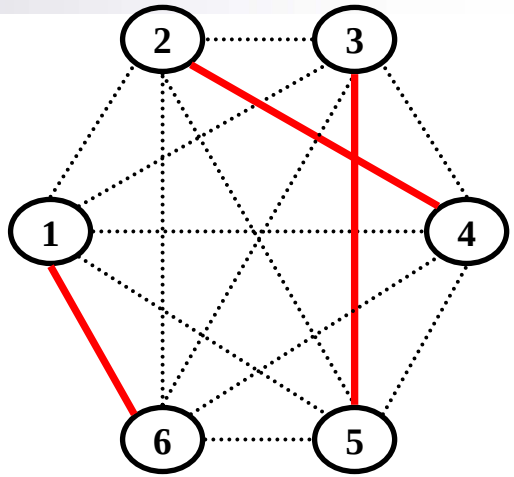
...

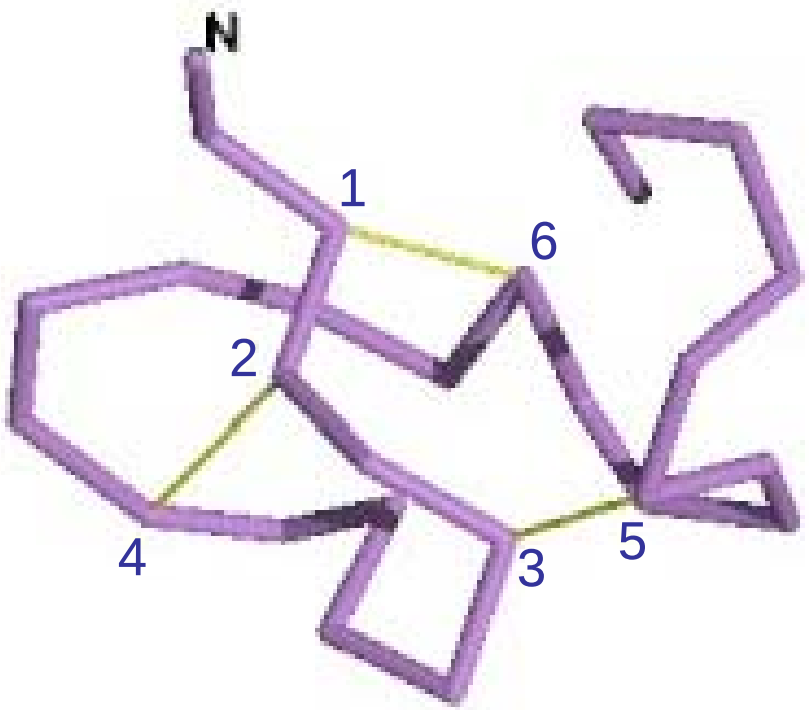
#(PP → IN NP)

...

#(NN → ‘sea’)

Disulfide bonds: non-bipartite matching





$$\mathbf{y} = (\underbrace{y_{12}, \ldots, y_{56}}_{n(n-1)/2})$$

w. features

$$s(\mathbf{x}, \mathbf{y}) = \sum_{ij} y_{ij} s_{ij}(\mathbf{x})$$

Fariselli & Casadio `01, Baldi et al. '04

Scoring function

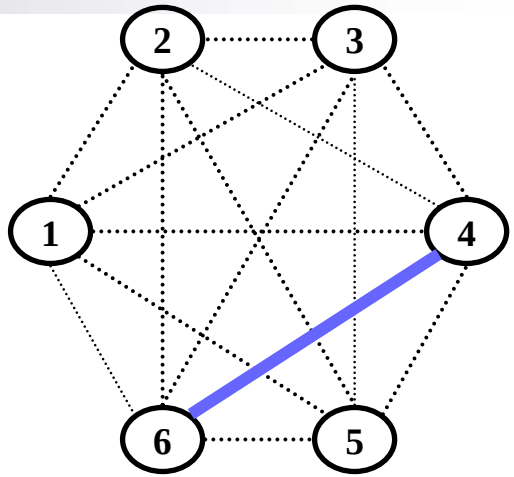
RSCCPCYWGGCPWGQNCYPEGCSGPKV

1 2 3 4 5 6

YWGGCPWGQ YPEGCSGPK

4 6

$\mathbf{x}_{46}$



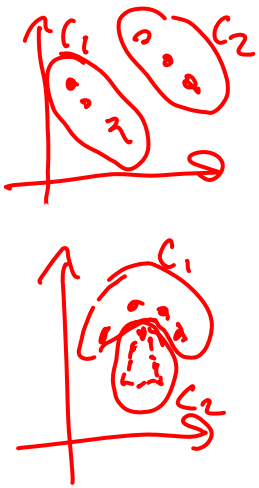
$$s_{46}(\mathbf{x}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{46})$$

String features:
residues, physical properties

$$s(\mathbf{x}, \mathbf{y}) = \sum_{ij} y_{ij} \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{ij}) \equiv \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$$

Learning to cluster

Input:
Solution to clustering problems



hand labeled clustering problems

→

Output:
Distance function

distance function

x_i, x_j

Score $s(x_i, x_j)$

$w^T f(x_i, x_j)$

Learning to cluster results

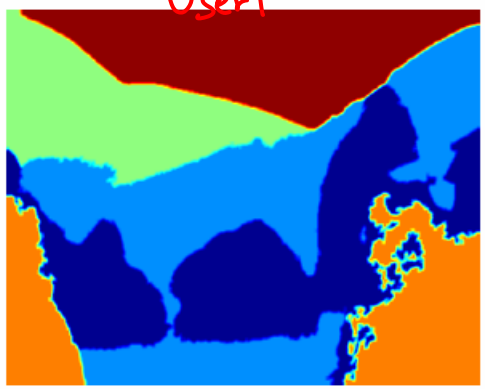
Input

User 1

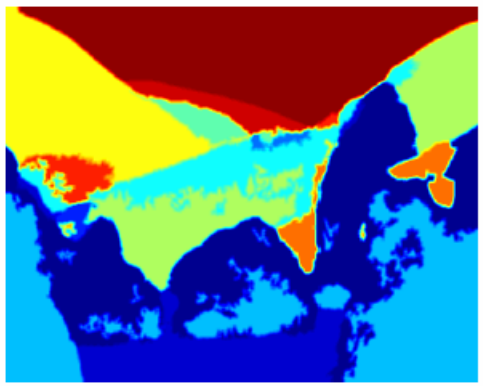
User 2

Output

Given image



Output 1



Output 2

Learning to optimize

- Given poly-time optimization problem
  - Minimum spanning tree
  - Bipartite matching
  - Shortest path
  - ...
- Max-margin learning optimization criterion
  - Weights of Markov network
  - Clustering distance function
  - Edge weights
  - ...

Conclusion

- Combine strengths of kernels and graphical models
  - Incorporate high-dim features
  - Exploit correlations and structure
- Efficient representation and learning procedure
  - Exact for triangulated networks (low-treewidth)
  - Approximate for untriangulated networks
  - Efficient SMO-like solver using network inference
- Generalization guarantees
  - Per-label bound
- Outperforms standard methods
  - OCR, Information Extraction and Hypertext Classification

Acknowledgements

- This lecture describes recent research (and slides) by Ben Taskar, more details:
  - **Ben Taskar's Thesis:** Learning Structured Prediction Models: A Large Margin Approach. Stanford University, CA, December 2004.
  - http://www.cs.berkeley.edu/~taskar/pubs/thesis.pdf