10-418 / 10-618 Machine Learning for Structured Data

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Midterm Exam Review

+

Structured Perceptron

+

Structured SVM

Matt Gormley
Lecture 14
Oct. 14, 2019

Reminders

- **Midterm Exam**
  - Thu, Oct. 17 at 6:30pm – 8:00pm
- **Homework 3: Structured SVM**
  - Out: Sat, Sep. 28
  - Due: Sat, Oct. 12 at 11:59pm

MIDTERM EXAM LOGISTICS

Midterm Exam

- **Time / Location**
  - **Time:** Evening Exam
    Thu, Oct. 17 at 6:30pm – 8:00pm
  - **Room:** Hamburg Hall A301
  - **Seats:** There will be **assigned seats**. Please arrive early to find yours.
  - Please watch Piazza carefully for announcements
- **Logistics**
  - Covered material: Lecture 1 – Lecture 13
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

Midterm Exam

- **Advice (for during the exam)**
  - Solve the easy problems first (e.g. multiple choice before derivations)
    - if a problem seems extremely complicated you're likely missing something
  - Don't leave any answer blank!
  - If you make an assumption, write it down
  - If you look at a question and don't know the answer:
    - we probably haven't told you the answer
    - but we've told you enough to work it out
    - imagine arguing for some answer and see if you like it

Topics for Midterm Exam

- Search-Based Structured Prediction
  - Reductions to Binary Classification
  - Learning to Search
  - RNN-LMs
  - seq2seq models
- Graphical Model Representation
  - Directed GMs vs. Undirected GMs vs. Factor Graphs
  - Bayesian Networks vs. Markov Random Fields vs. Conditional Random Fields
- Graphical Model Learning
  - Fully observed Bayesian Network learning
  - Fully observed MRF learning
  - Fully observed CRF learning
  - Parameterization of a GM
  - Neural potential functions
- Exact Inference
  - Three inference problems:
    (1) marginals
    (2) partition function
    (3) most probably assignment
  - Variable Elimination
  - Belief Propagation (sum-product and max-product)
  - MAP Inference via MILP

SAMPLE QUESTIONS

Sample Questions

Learning to Search

Suppose you are training a seq2seq model for supervised POS Tagging.

- Let the inputs to the encoder be $e_1, e_2, e_3, \ldots$
- Let the inputs to the decoder be $d_1, d_2, d_3, \ldots$
- Let the outputs of the decoder be $o_1, o_2, o_3, \ldots$

1. (1 point) **Short Answer**: Describe in words what the inputs to the encoder would be. Assume you are training with Teacher Forcing.

2. (1 point) **Short Answer**: Describe in words what the inputs of the decoder would be. Assume you are training with Teacher Forcing.

3. (1 point) **Short Answer**: Describe in words what the outputs of the decoder would be. Assume you are training with Teacher Forcing.

Sample Questions

Learning to Search

Suppose you are training a seq2seq model for supervised POS Tagging.

- Let the inputs to the encoder be e_1, e_2, e_3, …
- Let the inputs to the decoder be d_1, d_2, d_3, …
- Let the outputs of the decoder be o_1, o_2, o_3, …

4. (1 point) **Short Answer**: Describe in words what the inputs to the encoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

5. (1 point) **Short Answer**: Describe in words what the inputs of the decoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

6. (1 point) **Short Answer**: Describe in words what the outputs of the decoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

Sample Questions

6 Factor Graphs

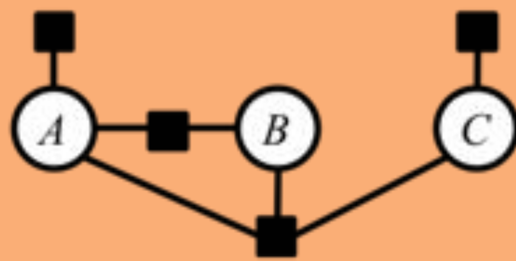


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

1. (2 points) **Short answer:** Consider the factor graph in Figure 4. Using the given factor names, write the partition function Z that ensures the joint probability distribution $p(a, b, c)$ sums-to-one.

Sample Questions

6 Factor Graphs

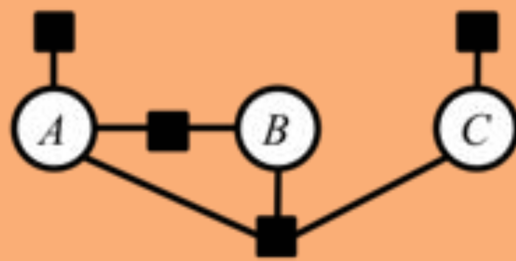


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

2. (2 points) **Short answer:** Using the given factor names, write the joint probability mass function $p(a, b, c)$ defined by the factor graph shown in Figure 4. *You may include the term Z directly in your answer—no need to copy it from above.*

Sample Questions

6 Factor Graphs

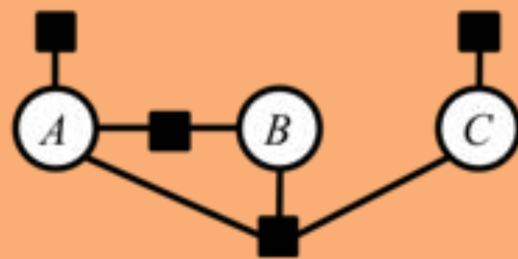


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

3. (2 points) **Drawing:** Suppose we have a joint probability distribution that factorizes as below:

$$p(w, x, y, z) \propto \psi_X(x)\psi_{X,Y}(x, y)\psi_{X,Y,Z}(x, y, z)\psi_{W,Z}(w, z)\psi_{Y,Z}(y, z)$$

where $\propto$ denotes *proportional to.* Draw the factor graph corresponding to this factorization of the joint distribution.

Sample Questions

6 Factor Graphs

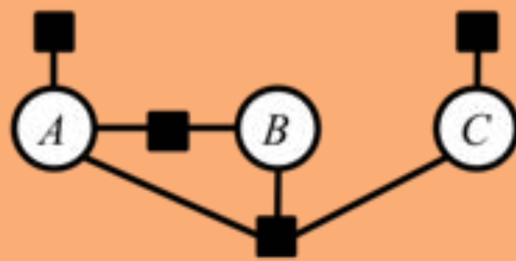


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0,1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a,b)$, $\psi_{A,B,C}(a,b,c)$, and $\psi_C(c)$.

3. (2 points) **Drawing:** Suppose we have a joint probability distribution that factorizes as below:

$$p(w,x,y,z) \propto \psi_X(x)\psi_{X,Y}(x,y)\psi_{X,Y,Z}(x,y,z)\psi_{W,Z}(w,z)\psi_{Y,Z}(y,z)$$

where $\propto$ denotes *proportional to.* Draw the factor graph corresponding to this factorization of the joint distribution.

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q,r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

1. (2 points) **Short answer:** Draw a table containing all values of the function $s(q,r) = \psi_Q(q)\psi_{Q,R}(q,r)$. *You may use the integer abbreviations: red=1, green=2, blue=3, pencil=1, crayon=2.*

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q, r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

2. (2 points) **Numerical answer:** What is the value of the partition function Z for the joint distribution $p(q, r)$?

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q, r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

3. (2 points) **Numerical answer:** What is the value of the joint probability $P(Q = green, R = crayon)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q,r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

4. (2 points) **Numerical answer:** What is the value of the marginal probability $P(Q = green)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q,r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

5. (2 points) **Short answer:** Suppose you run the Variable Elimination algorithm to eliminate the variable Q, resulting in a new factor graph with just one factor $m(r)$. Draw a table containing the values of this new factor.

Sample Questions

7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

Q	$\psi_Q(q)$
red	3
green	1
blue	2

Q	R	$\psi_{Q,R}(q,r)$
red	pencil	2
red	crayon	2
green	pencil	1
green	crayon	3
blue	pencil	4
blue	crayon	1

6. (2 points) **Numerical answer:** What is the value of the marginal probability $P(R = crayon)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

Sample Questions

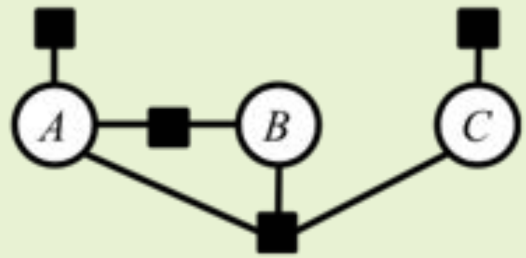


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

1. (1 point) **Drawing**: Suppose you are running the Variable Elimination algorithm. The first variable you eliminate is B. Draw the factor graph that results after you have eliminated variable B.

Sample Questions

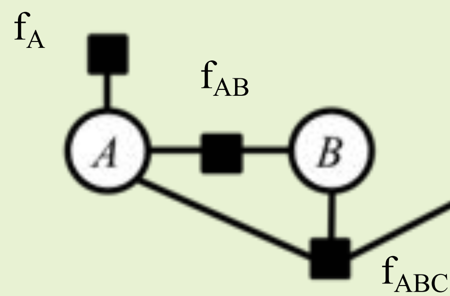
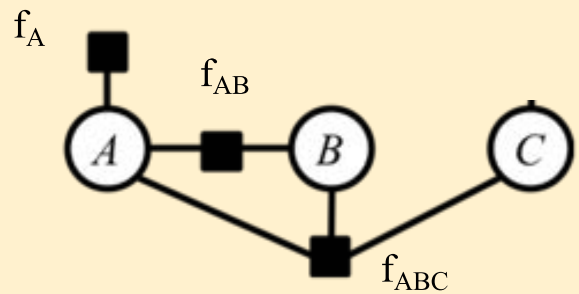


Figure 4: A factor graph over three binary random variables A, B, C, i.e. sampled values a, b, c from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

2. (1 point) **Numerical Answer**: Suppose you are running the Belief Propagation algorithm? How many messages are required to send a message from f_{ABC} to C?

Sample Questions



1. (1 point) Is there a Bayesian Network that would convert to the factor graph shown above? Is yes, draw an example of such a Bayesian Network. If not, explain why not.

Q&A

MAP INFERENCE AS MATHEMATICAL PROGRAMMING

Recall...

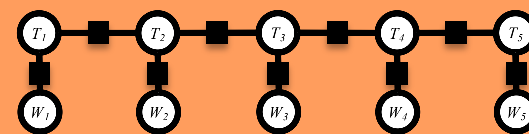
Exact Inference

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

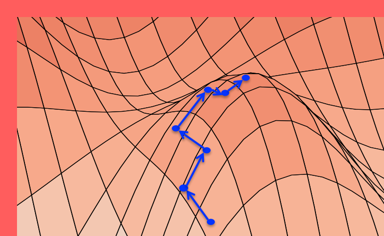
$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

3. MAP Inference

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

4. Learning

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



Recall...

5. Inference

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

1. Marginal Inference
Compute marginals of variables and cliques

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

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

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

5. Inference

Recall...

Three Tasks:

1. Marginal Inference
Compute marginals of variables and cliques

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

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

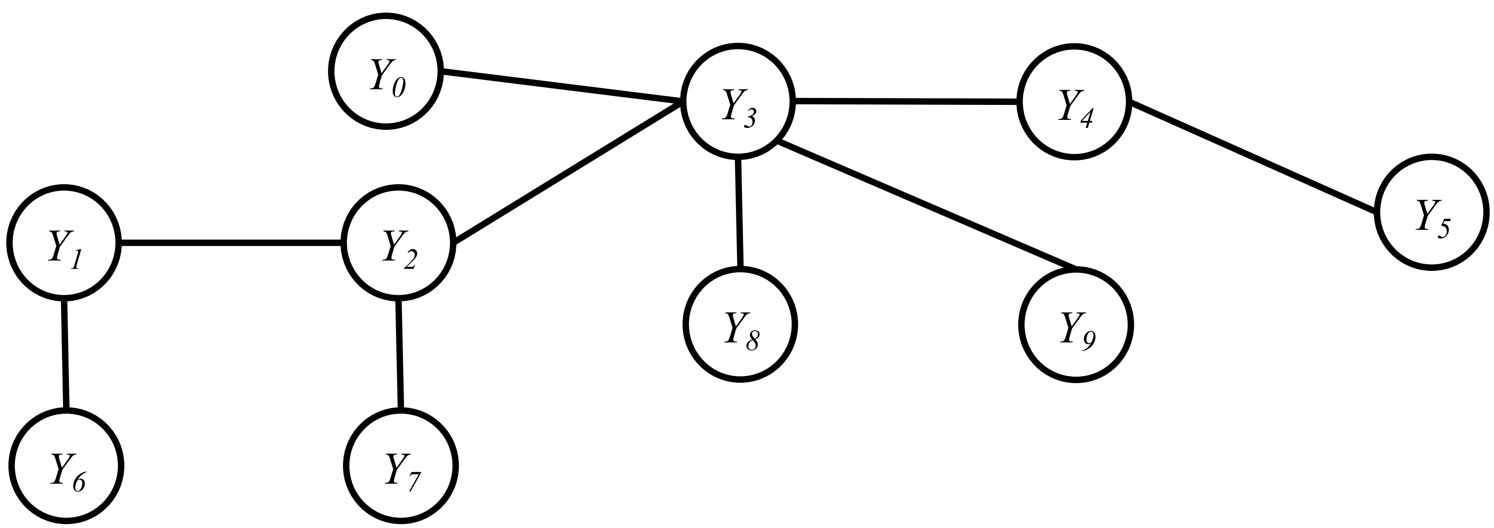
$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

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!

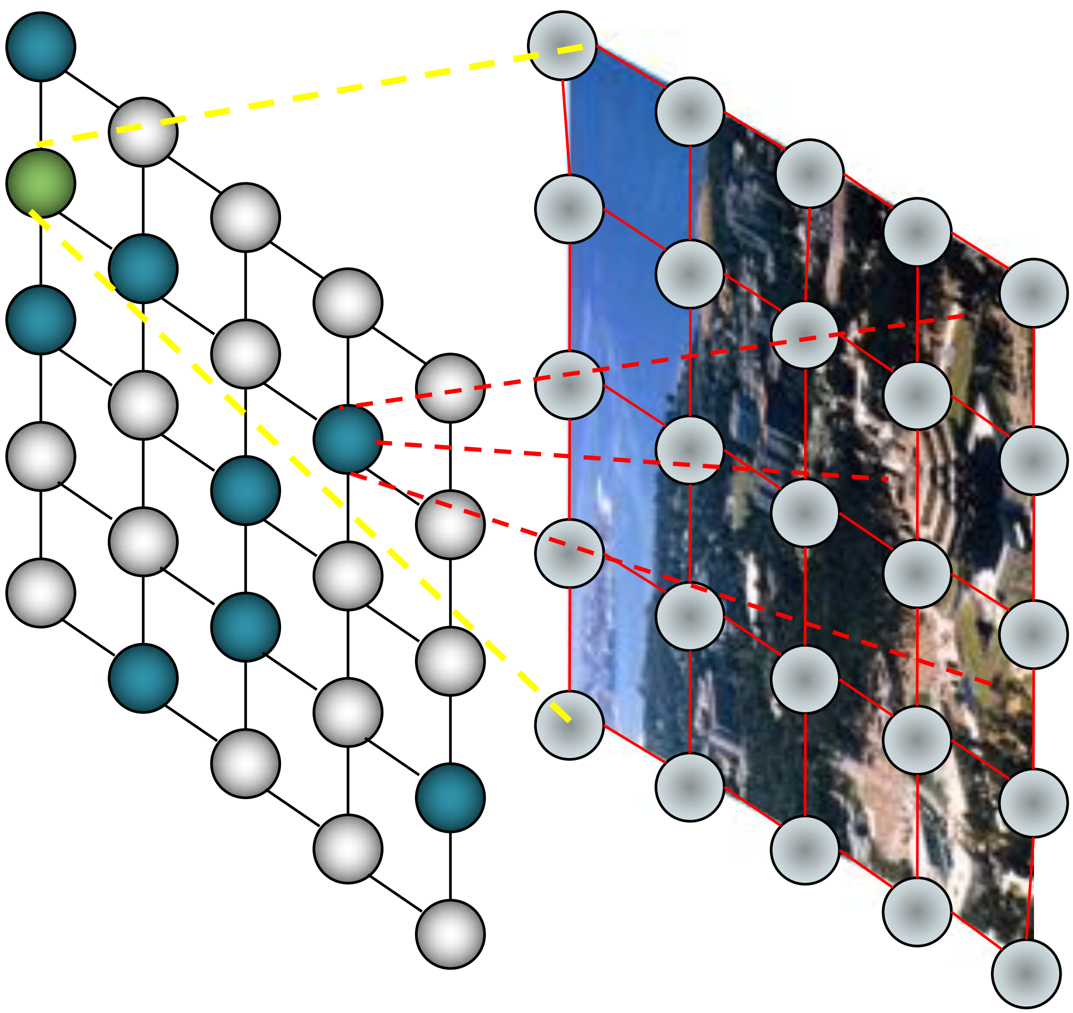
Integer Linear Programming

Whiteboard

– MAP Inference for a Binary Pairwise MRF as an ILP

– Question: What if we have non-binary variables?

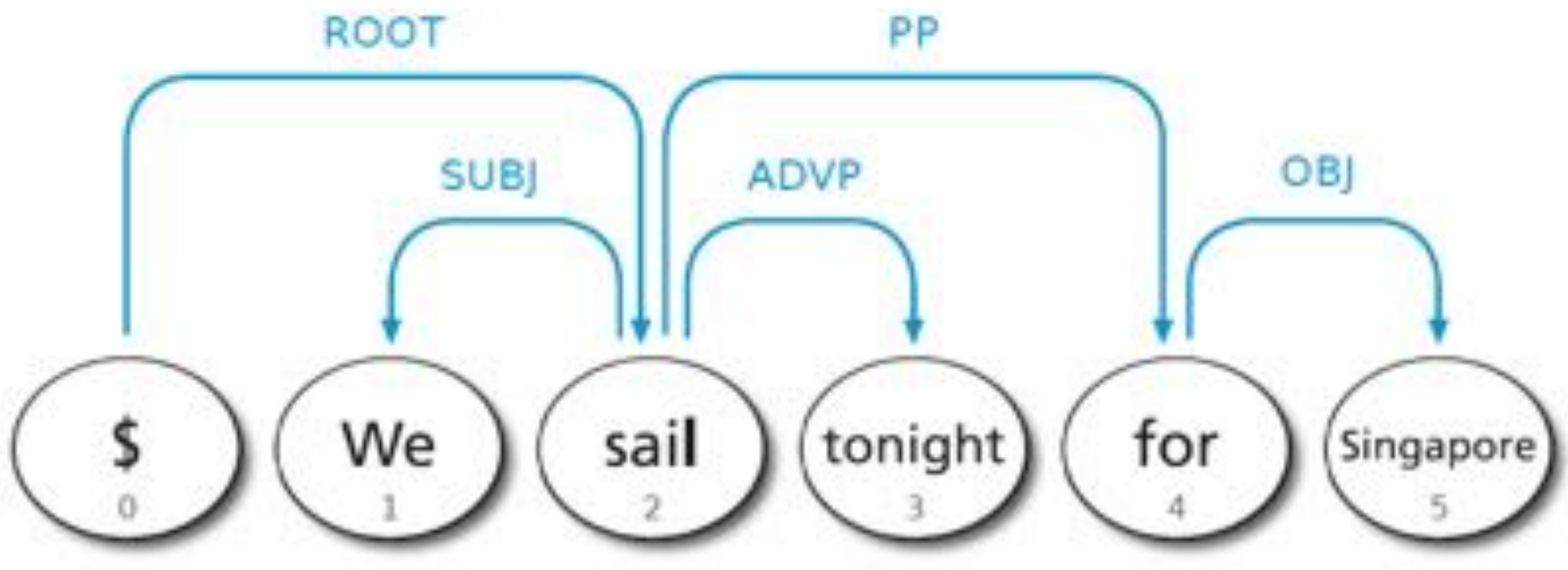
Image Segmentation



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

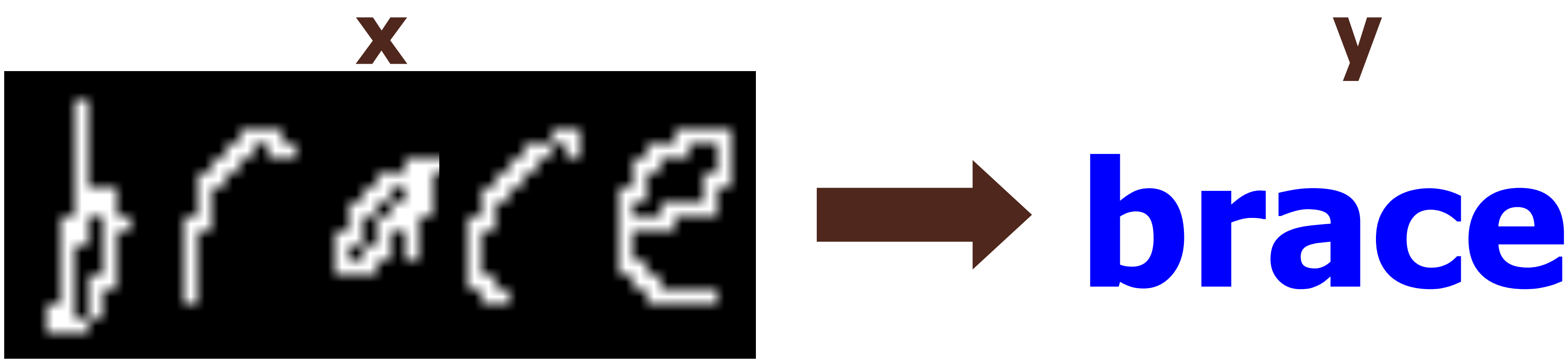
- Jointly segmenting/annotating images
- Image-image matching, image-text matching
- Problem:
  - Given structure (feature), learning $\vec{\theta}$
  - Learning sparse, interpretable, **predictive** structures/features

Dependency parsing of Sentences

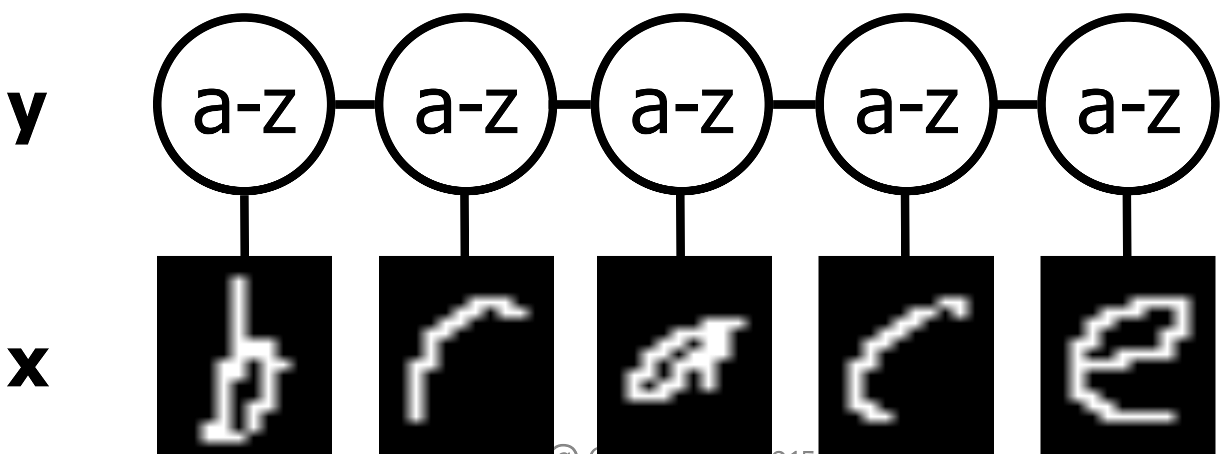


Challenge:
Structured outputs, and globally constrained to be a valid tree

OCR example



Sequential structure

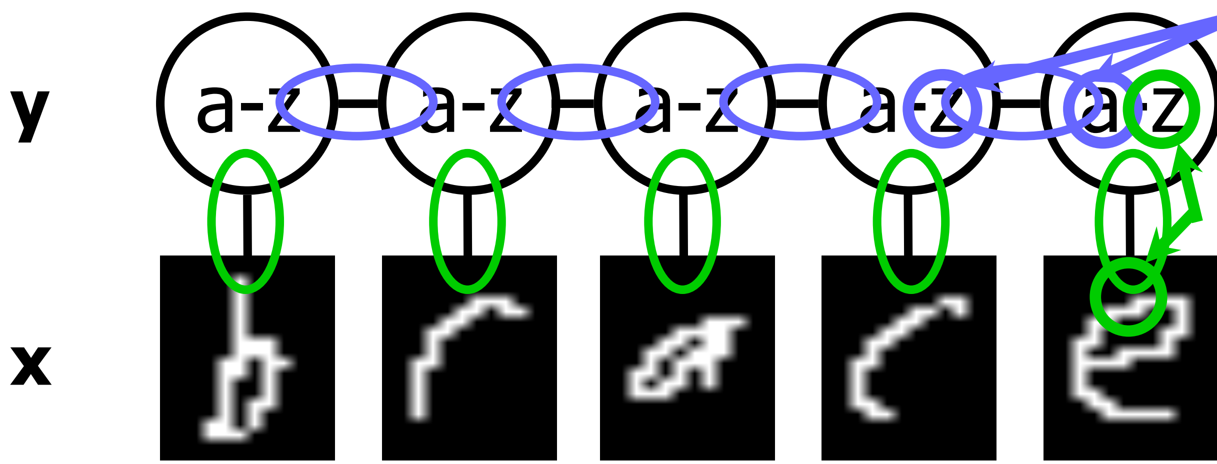


Linear-chain CRF for OCR

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

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

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



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

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

*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)

$$
\begin{array}{c}
\\
\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \\ : \\ \mathrm{Z} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_1(m) \\
\begin{bmatrix} 1 \\ 0 \\ : \\ 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_2(m) \\
\begin{bmatrix} 0 \\ 1 \\ : \\ 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_3(m) \\
\begin{bmatrix} 1 \\ 0 \\ : \\ 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_4(m) \\
\begin{bmatrix} 0 \\ 1 \\ : \\ 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_5(m) \\
\begin{bmatrix} 0 \\ 1 \\ : \\ 0 \end{bmatrix}
\end{array}
$$

$$
\begin{array}{c}
\\
\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \\ : \\ \mathrm{Z} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_{12}(m,n) \\
\begin{bmatrix} 0 & 1 & . & 0 \\ 0 & 0 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} \mathrm{A} & \mathrm{B} & . & \mathrm{Z} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_{23}(m,n) \\
\begin{bmatrix} 0 & 0 & . & 0 \\ 1 & 0 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} \mathrm{A} & \mathrm{B} & . & \mathrm{Z} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_{34}(m,n) \\
\begin{bmatrix} 0 & 1 & . & 0 \\ 0 & 0 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} \mathrm{A} & \mathrm{B} & . & \mathrm{Z} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
z_{45}(m,n) \\
\begin{bmatrix} 0 & 0 & . & 0 \\ 0 & 1 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\begin{bmatrix} \mathrm{A} & \mathrm{B} & . & \mathrm{Z} \end{bmatrix}
\end{array}
$$

$y \Rightarrow z$ 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:

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

$$z_k(n) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$z_j(m) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad z_{jk}(m,n) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

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

normalization $\sum_m z_j(m) = 1$

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

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

$\mathbf{y} \Rightarrow \mathbf{z}$ 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:

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

$$z_k(n): \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$z_j(m): \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad z_{jk}(m,n): \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left.\begin{aligned} & z_j(m) \geq 0; \; z_{jk}(m,n) \geq 0; \\ \text{normalization} \quad & \sum_m z_j(m) = 1 \\ \text{agreement} \quad & \sum_n z_{jk}(m,n) = z_j(m) \end{aligned}\right\} \mathbf{Az} = \mathbf{b}$$

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

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!

STRUCTURED PERCEPTRON

Structured Perceptron

Whiteboard

- Multiclass Perceptron
- Structured Perceptron
- Structured Perceptron with Averaging
- Definition: Margin for Structured Outputs
- Mistake Bound for Structured Perceptron

Structured Perceptron

Mistake Bound:

Definition 1 *Let* $\overline{\mathbf{GEN}}(x_i) = \mathbf{GEN}(x_i) - \{y_i\}$. *In other words* $\overline{\mathbf{GEN}}(x_i)$ *is the set of* incorrect *candidates for an example* x_i. *We will say that a training sequence* (x_i, y_i) *for* $i = 1 \ldots n$ *is* **separable with margin** $\delta > 0$ *if there exists some vector* $\mathbf{U}$ *with* $||\mathbf{U}|| = 1$ *such that*

$$\forall i, \forall z \in \overline{\mathbf{GEN}}(x_i), \quad \mathbf{U} \cdot \Phi(x_i, y_i) - \mathbf{U} \cdot \Phi(x_i, z) \geq \delta \quad (3)$$

($||\mathbf{U}||$ *is the 2-norm of* $\mathbf{U}$, *i.e.,* $||\mathbf{U}|| = \sqrt{\sum_s \mathbf{U}_s^2}$.*)*

Theorem 1 *For any training sequence* (x_i, y_i) *which is separable with margin* δ, *then for the perceptron algorithm in figure 2*

$$\textit{Number of mistakes} \leq \frac{R^2}{\delta^2}$$

where R *is a constant such that* $\forall i, \forall z \in \overline{\mathbf{GEN}}(x_i)$ $||\Phi(x_i, y_i) - \Phi(x_i, z)|| \leq R$.

from Collins (2002)

Structured Perceptron

- Results from Collins (2002) on two **sequence tagging** problems
- Metrics:
  - **F-measure:** higher is better
  - **Error:** lower is better
- Comparison of…
  - Structured Perceptron **with** and **without** averaging
  - Maximum entropy Markov model (**ME**MM)
- Takeaways:
  - incredibly **easy to implement**
  - typically **blazing fast**

NP Chunking Results

Method	F-Measure	Numits
Perc, avg, cc=0	93.53	13
Perc, noavg, cc=0	93.04	35
Perc, avg, cc=5	93.33	9
Perc, noavg, cc=5	91.88	39
ME, cc=0	92.34	900
ME, cc=5	92.65	200

POS Tagging Results

Method	Error rate/%	Numits
Perc, avg, cc=0	2.93	10
Perc, noavg, cc=0	3.68	20
Perc, avg, cc=5	3.03	6
Perc, noavg, cc=5	4.04	17
ME, cc=0	3.4	100
ME, cc=5	3.28	200

Figure 4: Results for various methods on the part-of-speech tagging and chunking tasks on development data. All scores are error percentages. Numits is the number of training iterations at which the best score is achieved. Perc is the perceptron algorithm, ME is the maximum entropy method. Avg/noavg is the perceptron with or without averaged parameter vectors. cc=5 means only features occurring 5 times or more in training are included, cc=0 means all features in training are included.

Table from Collins (2002)

aka. Max-Margin Markov Networks (M^3Ns)

STRUCTURED SVM

Structured Perceptron

Whiteboard

- Warmup: Binary SVM
- Warmup: Binary SVM Hinge Loss
- Structured Large Margin
- Structured Hinge Loss
- Gradient of Structured Hinge Loss
- SGD for Structured SVM
- Loss Augmented MAP Inference

Max vs "Soft-Max" Margin

- SVMs:

$$\min_{\mathbf{w}} k||\mathbf{w}||^2 - \sum_i \underbrace{\left(\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}^i) - \max_{\mathbf{y}} \left(\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}) + \ell_i(y)\right)\right)}_{\text{Hard (Penalized) Margin}}$$

- Maxent:

$$\min_{\mathbf{w}} \; k||w||^2 - \sum_i \underbrace{\left(\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}^i) - \log \sum_{\mathbf{y}} \exp\left(\mathbf{w}^\top \mathbf{f}_i(\mathbf{y})\right)\right)}_{\text{Soft Margin}}$$

- Very similar! Both try to make the true score better than a function of the other scores.
  - The SVM tries to beat the augmented runner-up
  - The maxent classifier tries to beat the "soft-max"

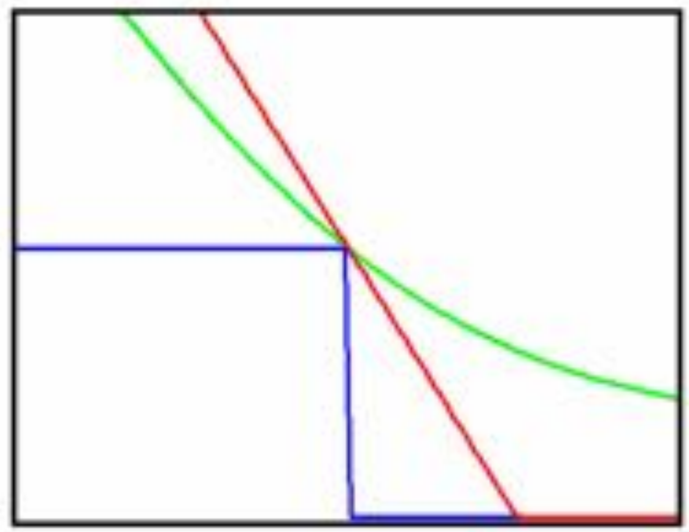
Slide from Klein & Taskar (ACL 2005 Tutorial)

Hinge Loss

- Consider the per-instance SVM objective:

$$\min_{\mathbf{w}} k||\mathbf{w}||^2 - \sum_i \left(\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}^i) - \max_{\mathbf{y}} \left[\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}) + \ell_i(y) \right] \right)$$

- This is called the "hinge loss"
  - Upper bounds zero-one loss
  - Unlike maxent / log loss, you stop gaining objective once the true label wins by enough
  - You can start from here and derive the SVM objective



Slide from Klein & Taskar (ACL 2005 Tutorial)

Max (Conditional) Likelihood

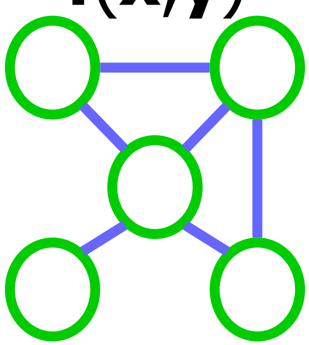
D

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

…

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

f(x,y)



Estimation

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

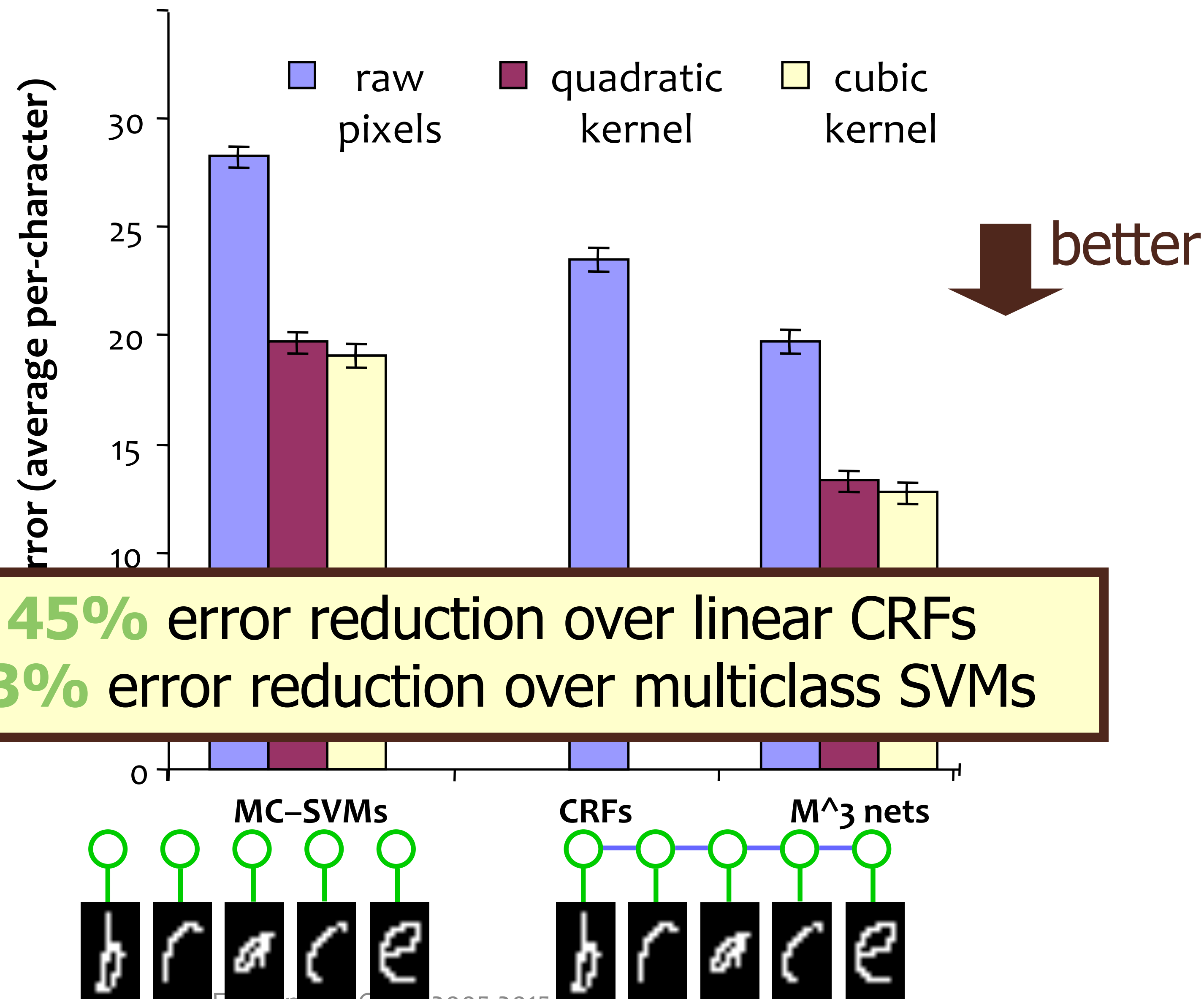
10-fold Train/Test

5000/50000 letters

600/6000 words

Models:

- Multiclass-SVMs*
- CRFs
- M^3 nets



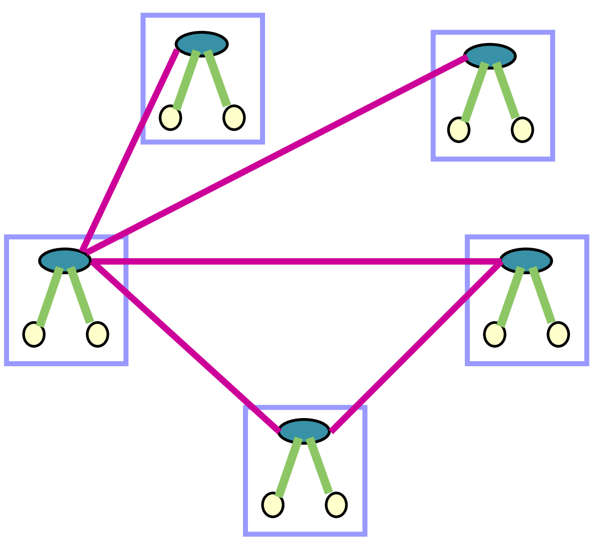
45% error reduction over linear CRFs

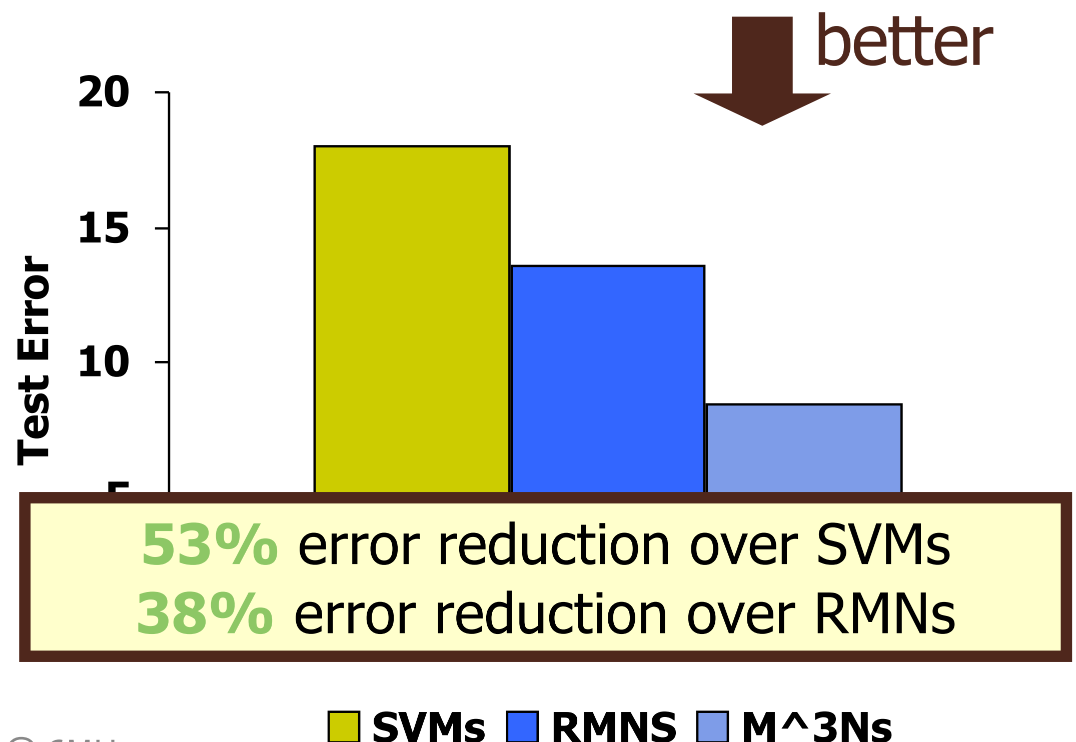
33% error reduction over multiclass SVMs

*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



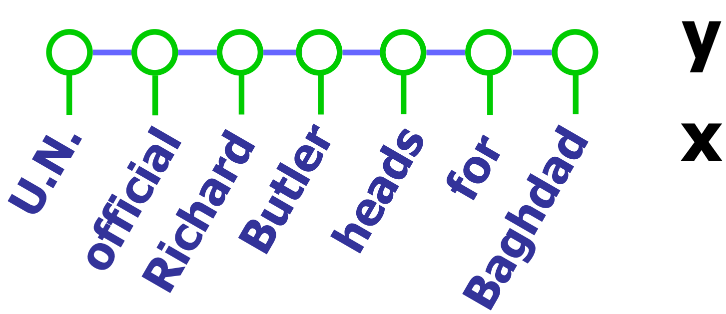


53% error reduction over SVMs
38% error reduction over RMNs

*Taskar et al 02

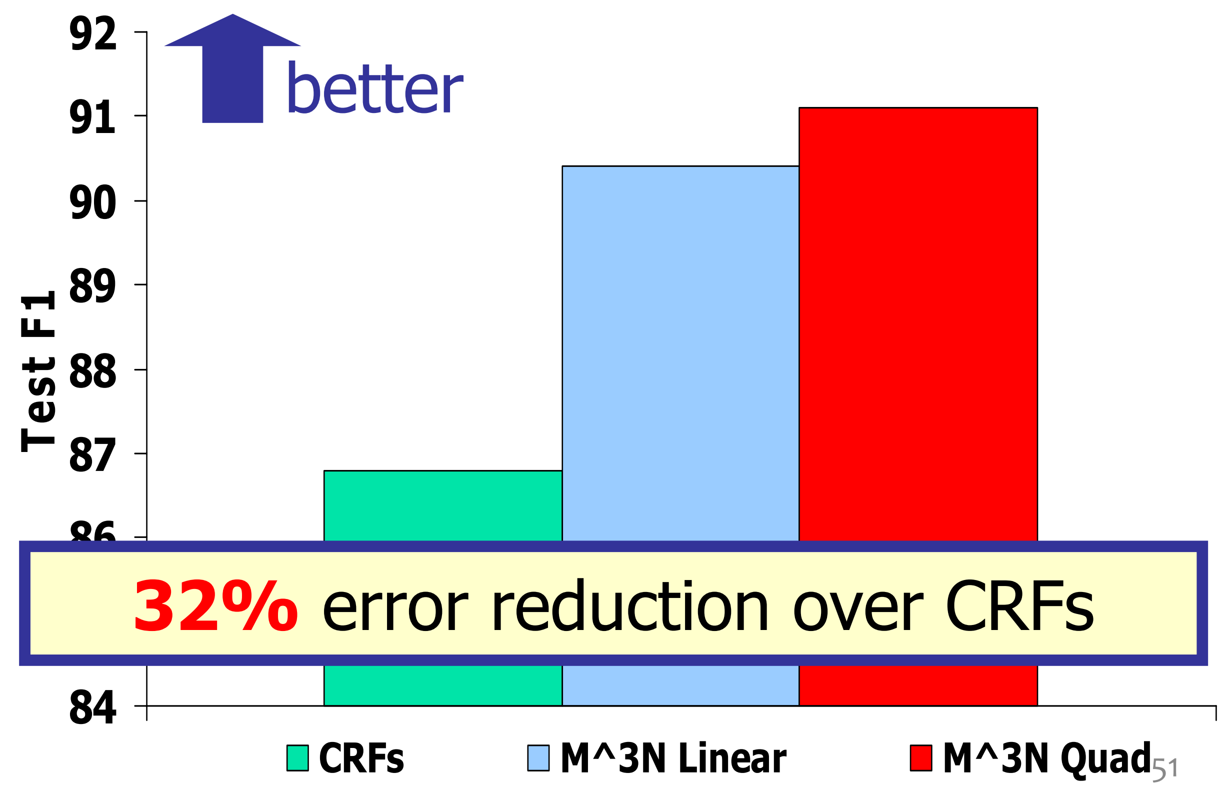
Named Entity Recognition

- 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

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

Associative Markov networks

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

spin-images, point height

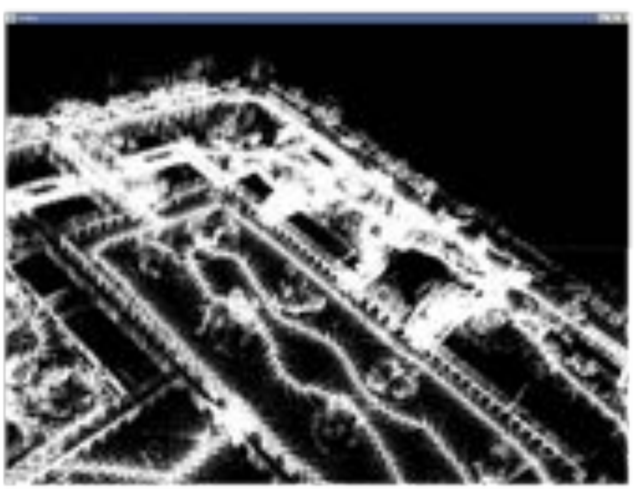
length of edge, edge orientation

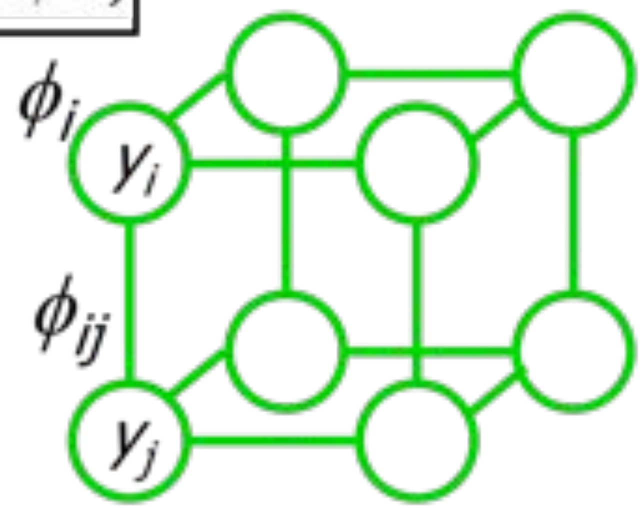
"associative" restriction

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

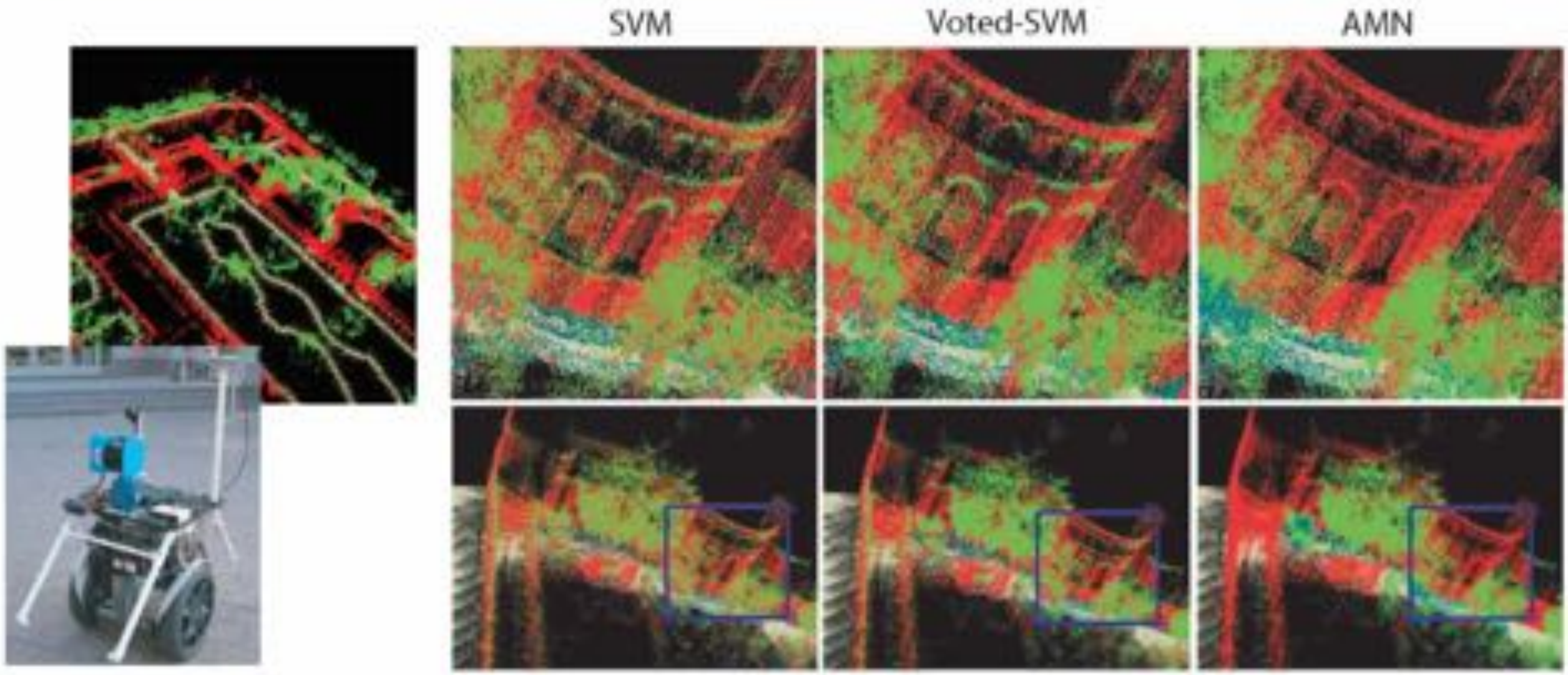
bonus

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





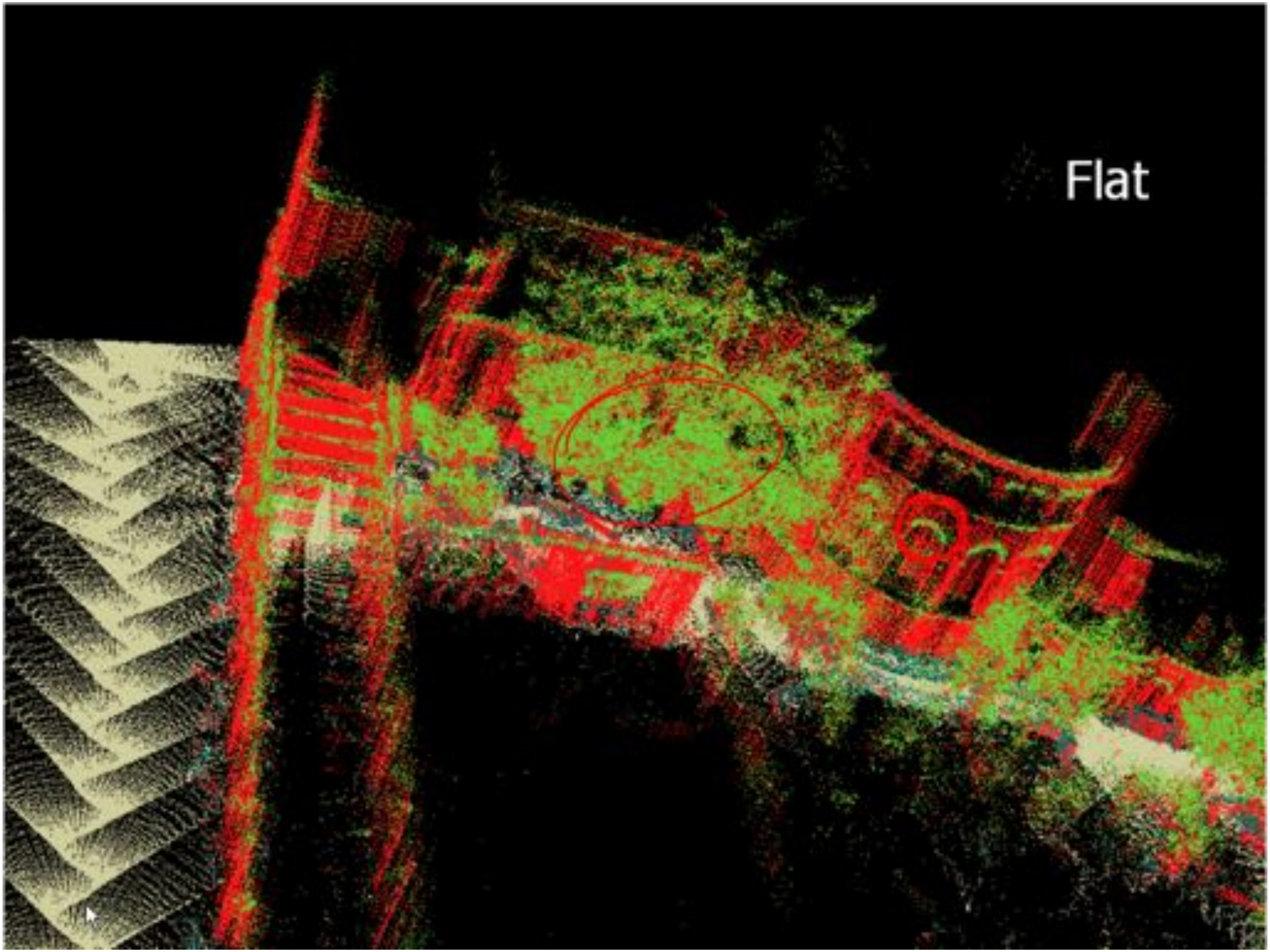
Max-margin AMNs results



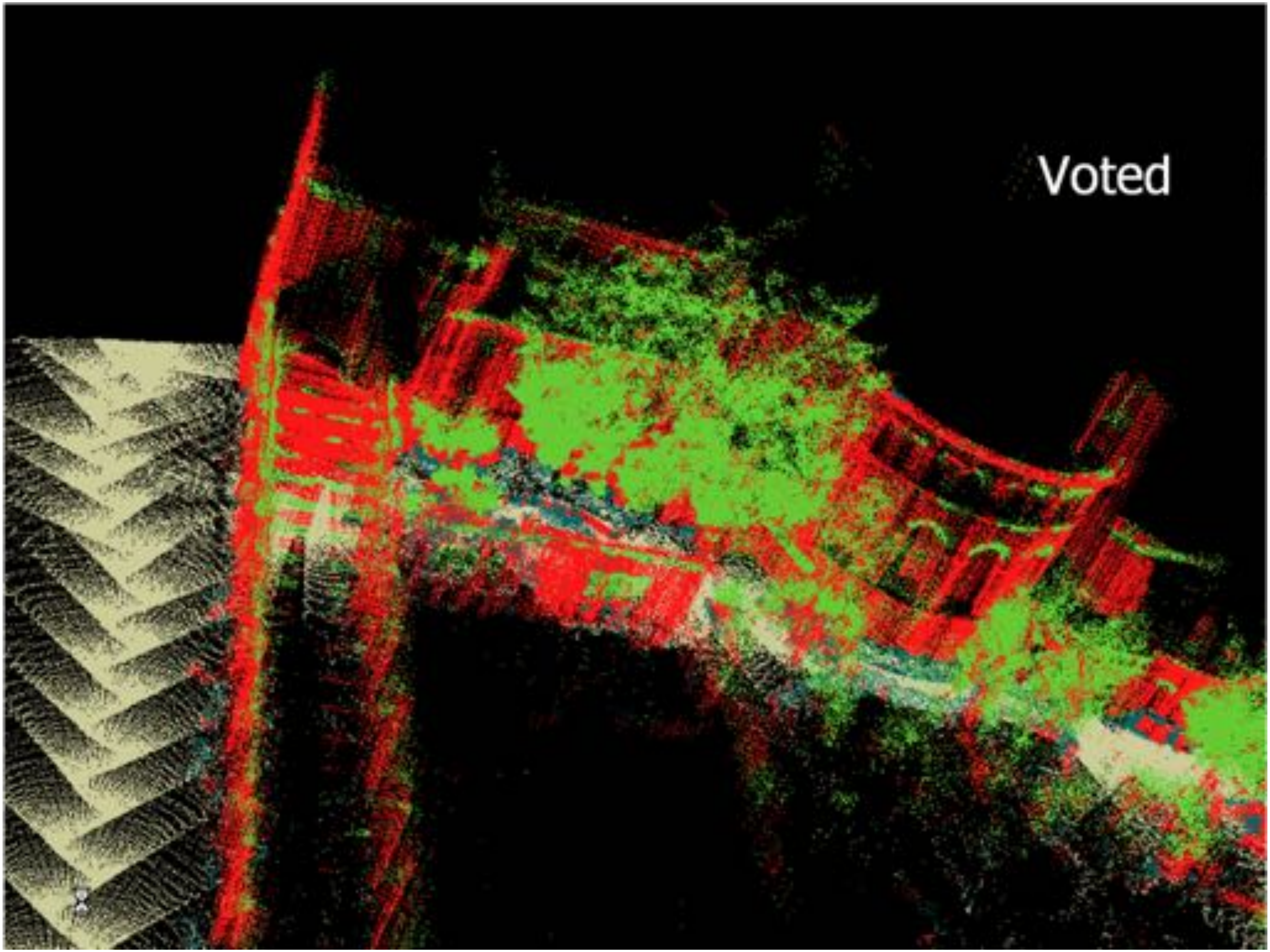
Label: ground, building, tree, shrub

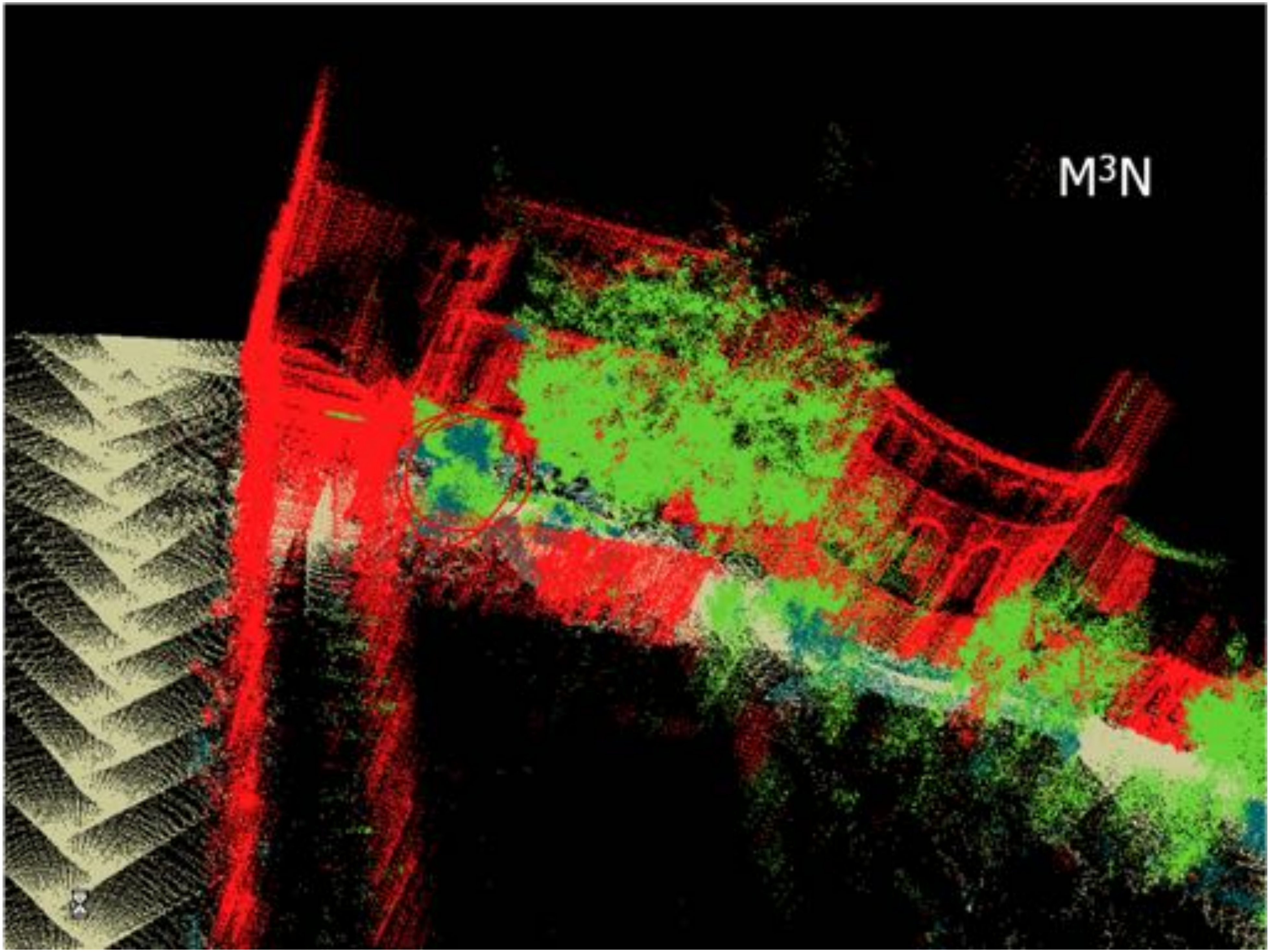
Training: 30 thousand points   Testing: 3 million points

Slide from Guestrin, 10-701, 2005



Slide from Guestrin, 10-701, 2005




M³N

Segmentation results

Hand labeled 180K test points

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