

---

# Tutorial on Conditional Random Fields for Sequence Prediction

---

Ariadna Quattoni

---

# RoadMap

- Sequence Prediction Problem
  - CRFs for Sequence Prediction
  - Generalizations of CRFs
  - Hidden Conditional Random Fields (HCRFs)
  - HCRFs for Object Recognition
-

---

# RoadMap

- **Sequence Prediction Problem**
  - CRFs for Sequence Prediction
  - Generalizations of CRFs
  - Hidden Conditional Random Fields (HCRFs)
  - HCRFs for Object Recognition
-

# Sequence Prediction Problem

## Example: Part-of-Speech Tagging

**X**



$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9]$

He reckons the current account deficit will narrow significantly

**Y**



[PRP] [VB] [DT] [JJ] [NN] [NN] [MD] [VB] [RB]

$[y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8 \quad y_9]$

---

## Gesture Recognition

**X**



**Y**



[HTF] [HTF] [HTF] [HOF] [HOF] [HOS]

---

---

# RoadMap

- Sequence Prediction Problem
  - **CRFs for Sequence Prediction**
  - Generalizations of CRFs
  - Hidden Conditional Random Fields (HCRFs)
  - HCRFs for Object Recognition
-

---

## Conditional Random Fields: Modelling the Conditional Distribution

Model the Conditional Distribution:

$$P(\mathbf{y} \mid \mathbf{x})$$

To predict a sequence compute:

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x})$$

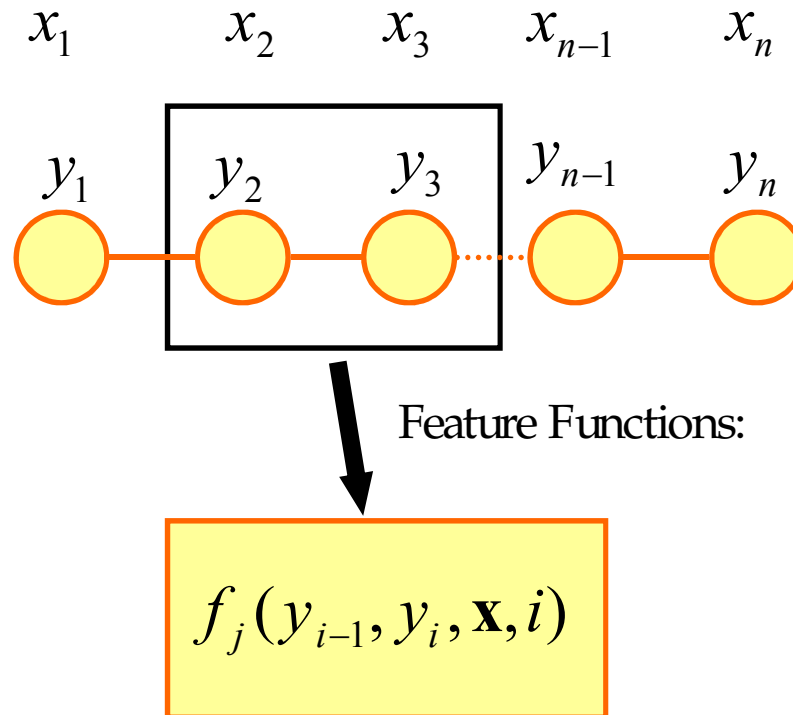


Must be able to compute it efficiently.

---

---

# Conditional Random Fields: Feature Functions



## Feature Functions

Express some characteristic of the empirical distribution that we wish to hold in the model distribution

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

1 if  $y_{i-1} = IN$  and  
 $y_i = NNP$  and  
 $x_i = September$

0 otherwise

---

## Conditional Random Fields:: Distribution

Label sequence modelled as a normalized product of feature functions:

$$P(\mathbf{y} | \mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i)$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i)$$

The model is log-linear on the Feature Functions

---

## Parameter Estimation: Maximum Likelihood

IID training samples:

$$D = [(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \dots, (\mathbf{x}^m, y^m)]$$

(negative) Conditional Log-Likelihood:

$$\begin{aligned} L(\lambda, D) &= -\log \left( \prod_{k=1}^m P(y^k | \mathbf{x}^k, \lambda) \right) \\ &= -\sum_{k=1}^m \log \left[ \frac{1}{Z(\mathbf{x}_m)} \exp \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}^k, y_i^k, \mathbf{x}^m, i) \right] \end{aligned}$$

---

## Parameter Estimation: Maximum Likelihood

### Maximum Likelihood Estimation

Set optimal parameters to be:

$$\lambda^* = \arg \min_{\lambda} L(\lambda, D) + C \frac{1}{2} \|\lambda\|^2$$

This function is convex, i.e. no local minimums

---

# Parameter Estimation: Optimization

$$\text{Let: } F_j(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^n f_j(y_{i-1}, y_i, \mathbf{x}, i)$$

Differentiating the log-likelihood with respect to parameter  $\lambda_j$

$$\frac{\partial L(\lambda, D)}{\partial \lambda_j} = \frac{-1}{m} \sum_{k=1}^m F_j(\mathbf{y}^k, \mathbf{x}^k) + \sum_{k=1}^m E_{P(\mathbf{y}|\mathbf{x}^k, \lambda)} [F_j(\mathbf{y}, \mathbf{x}^k)]$$

Observed Mean  
Feature Value

Expected Feature  
Value Under  
The Model

---

# Parameter Estimation: Optimization

Generally, it is not possible to find an analytic solution to the previous objective.

Iterative techniques, i.e. gradient based methods.

---

## Maximum Entropy Interpretation

Notice that at the optimal solution of:

$$\lambda^* = \arg \min_{\lambda} L(\lambda, D) + C \frac{1}{2} \|\lambda\|^2$$

We must have that:

$$\frac{1}{m} \sum_{k=1}^m F_j(\mathbf{y}^k, \mathbf{x}^k) = \sum_{k=1}^m E_{P(\mathbf{y}|\mathbf{x}^k, \lambda)} [F_j(\mathbf{y}, \mathbf{x}^k)]$$

Maximizing log-likelihood  $\approx$

Finding max-entropy distribution that satisfies the set of constraints defined by the feature functions

## CRF's Inference

Given a model, i.e. parameter values

Can we compute the following efficiently?

Best Label  
Sequence

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y} | \mathbf{x}, \boldsymbol{\lambda}^*)$$

Expected  
Values

$$\begin{aligned} \sum_{k=1}^m E_{P(\mathbf{y}|\mathbf{x}^k, \boldsymbol{\lambda})} [F_j(\mathbf{y}, \mathbf{x}^k)] &= \sum_{k=1}^m \sum_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}^k, \boldsymbol{\lambda}) F_j(\mathbf{y}, \mathbf{x}^k) \\ &= \sum_{k=1}^m \sum_{i=1}^n \sum_{\mathbf{y}: [y_{i-1}=a, y_i=b]} p(y_{i-1} = a, y_i = b | \mathbf{x}^k, \boldsymbol{\lambda}) f_j(a, b, \mathbf{x}^k, i) \end{aligned}$$

Both can be computed using dynamic programming.

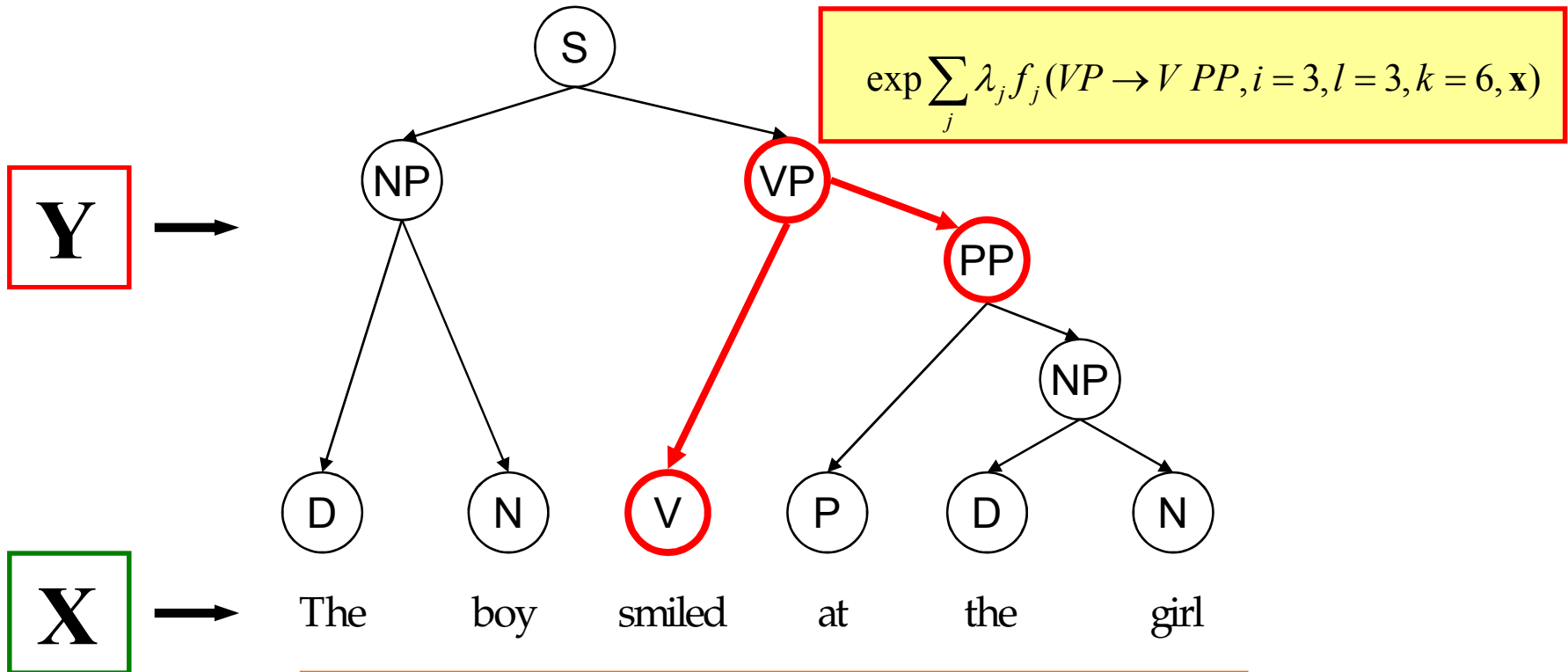
---

# RoadMap

- Sequence Prediction Problem
  - CRFs for Sequence Prediction
  - **Generalizations of CRFs**
  - Hidden Conditional Random Fields (HCRFs)
  - HCRFs for Object Recognition
-

# Generalization I: CRFs Beyond Sequences

Predicting Trees: Application Constituent Parsing



$$P(\mathbf{y} | \mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{\langle A \rightarrow BC, i, j, k \rangle \in \mathbf{y}} \sum_j \lambda_j f_j(A \rightarrow BC, i, l, k, \mathbf{x})$$

---

## Generalization II: Factorized Linear Models

To predict a sequence compute:

$$\begin{aligned} \mathbf{y}^* &= \arg \max_{\mathbf{y}} \frac{1}{Z(\mathbf{x})} \exp \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i) \\ &= \arg \max_{\mathbf{y}} \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i) \end{aligned} \quad \longrightarrow \text{Linear Model}$$

Objective: making accurate predictions on unseen data

The parameters of the linear model can be optimized using other loss functions

---

## Generalization II: Factorized Linear Models

### Structured Hinge Loss

Let  $\mathbf{z}$  be the correct label sequence:

$$l(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = \begin{cases} 0 & \text{if } \sum_{i=1}^n \sum_j \lambda_j f_j(\mathbf{z}_{i-1}, \mathbf{z}_i, \mathbf{x}, i) > \arg \max_{\mathbf{y} \neq \mathbf{z}} \sum_{i=1}^n \sum_j \lambda_j f_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) + 1 \\ \arg \max_{\mathbf{y} \neq \mathbf{z}} \sum_{i=1}^n \sum_j \lambda_j f_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) - \sum_{i=1}^n \sum_j \lambda_j f_j(\mathbf{z}_{i-1}, \mathbf{z}_i, \mathbf{x}, i) - 1 & \text{otherwise} \end{cases}$$

### Structured SVM

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \sum_{k=1}^m l(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}) + C \frac{1}{2} \|\boldsymbol{\lambda}\|^2$$

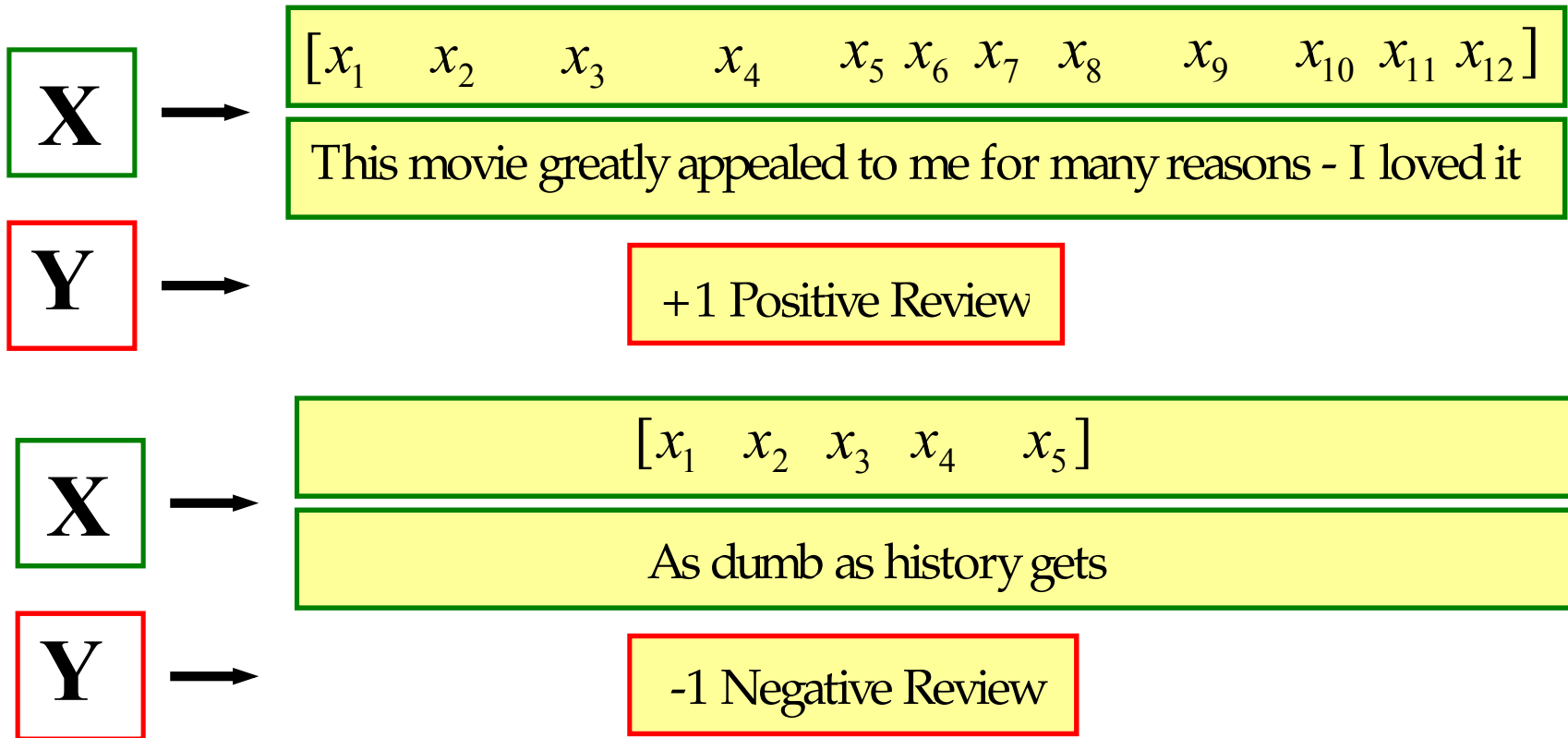
---

# RoadMap

- Sequence Prediction Problem
  - CRFs for Sequence Prediction
  - Generalizations of CRFs
  - **Hidden Conditional Random Fields (HCRFs)**
  - HCRFs for Object Recognition
-

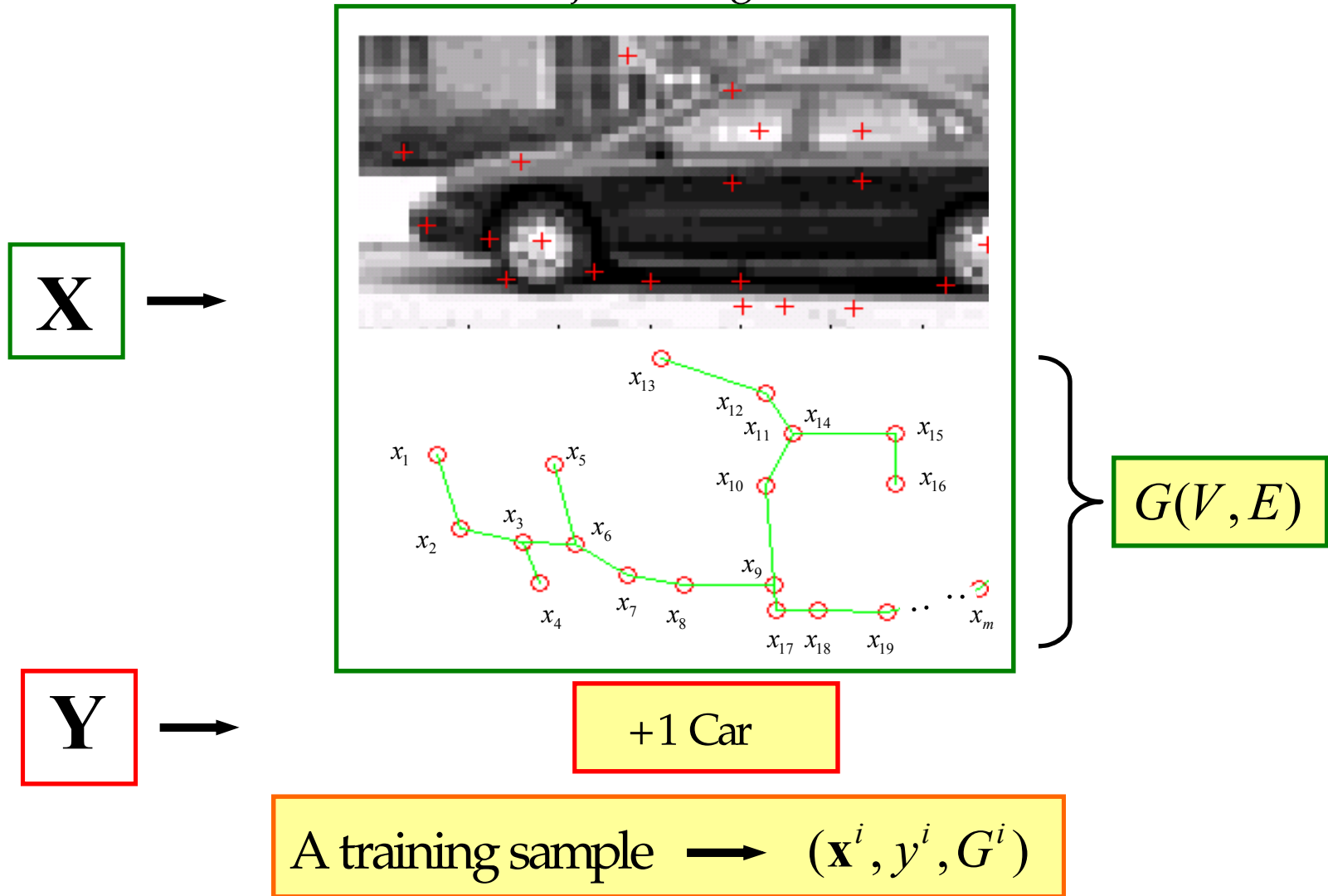
# Hidden Conditional Random Fields

## Sentiment Detection



# Hidden Conditional Random Fields

## Object Recognition



## Hidden Conditional Random Fields

Model the conditional probability:  $P(y | \mathbf{x}, G)$

We introduce hidden variables:  $\mathbf{h} = \{h_1, h_2, \dots, h_m\}$   $h \in H$

Analogous to the standard CRF we define:

$$P(y, \mathbf{h} | \mathbf{x}, G, \lambda) = \frac{\exp^{\psi(y, \mathbf{h}, \mathbf{x}, G, \lambda)}}{\sum_{y', \mathbf{h}} \exp^{\psi(y', \mathbf{h}, \mathbf{x}, G, \lambda)}}$$

$$P(y | \mathbf{x}, G, \lambda) = \sum_{\mathbf{h}} P(y, \mathbf{h} | \mathbf{x}, G, \lambda) = \frac{\sum_{\mathbf{h}} \exp^{\psi(y, \mathbf{h}, \mathbf{x}, \lambda)}}{\sum_{\mathbf{h}, y'} \exp^{\psi(y', \mathbf{h}, \mathbf{x}, \lambda)}}$$

$\psi(y, \mathbf{h}, \mathbf{x}, G, \lambda)$  Maps a configuration to the reals.

---

# Hidden Conditional Random Fields

## Feature Functions

$$\psi(y, \mathbf{h}, \mathbf{x}, G, \lambda) = \sum_{k \in V} \sum_j \lambda_j^1 f_j^1(k, y, h_k, \mathbf{x}) + \sum_{(k,l) \in E} \sum_j \lambda_j^2 f_j^2(k, l, y, h_k, h_l, \mathbf{x})$$

---

## Parameter Estimation

Maximum Likelihood:

$$L(\boldsymbol{\lambda}, D) = -\sum_i \log P(y^i | \mathbf{x}^i, G^i, \boldsymbol{\lambda})$$

Find optimal parameters:

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda}, D) + C \frac{1}{2} \|\boldsymbol{\lambda}\|^2$$

Iterative techniques, i.e. gradient based methods.  
But now the function is not convex!!!

At test time make prediction:

$$y^* = \arg \max_y P(y | \mathbf{x}, G, \boldsymbol{\lambda}^*)$$

## Parameter Estimation

The derivative of the loss function

is given by:  $\frac{\partial L_i(\mathbf{x}^i, G^i, y)}{\partial \lambda_j^1}$

$$- \sum_{y \in Y, k \in V^i, a \in H} P(h_k = a, y | \mathbf{x}^i, G^i, \lambda) f_j^1(k, y, a, \mathbf{x}^i) + \sum_{k \in V^i, a \in H} P(h_k = a | y^i, \mathbf{x}^i, G^i, \lambda) f_j^1(k, y^i, a, \mathbf{x}^i)$$

The derivative can be expressed in terms of components:

$$P(h_j = a | \mathbf{x}, G, \lambda) \quad P(h_k = a, h_l = b | \mathbf{x}, G, \lambda) \quad P(y | \mathbf{x}, G, \lambda)$$

that can be calculated using dynamic programming.  
Similarly the argmax can also be computed efficiently.

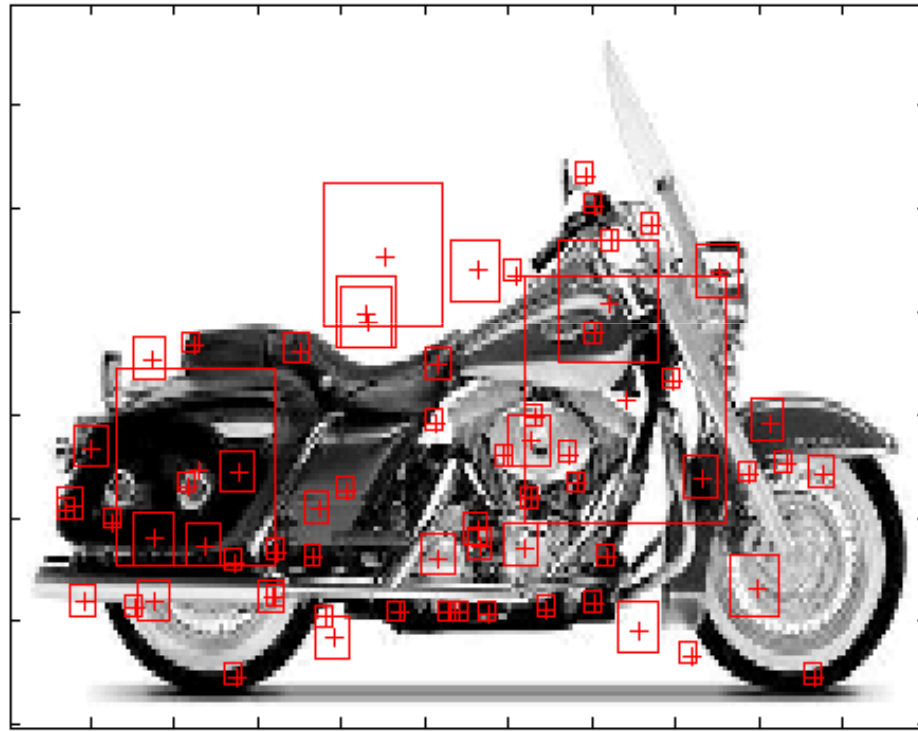
---

# RoadMap

- Sequence Prediction Problem
  - CRFs for Sequence Prediction
  - Generalizations of CRFs
  - Hidden Conditional Random Fields (HCRFs)
  - **HCRFs for Object Recognition**
-

# Application :: Object Recognition

## SemiSupervised Part-based Models



$$\mathbf{X} = \{x_1, \dots, x_m\}$$

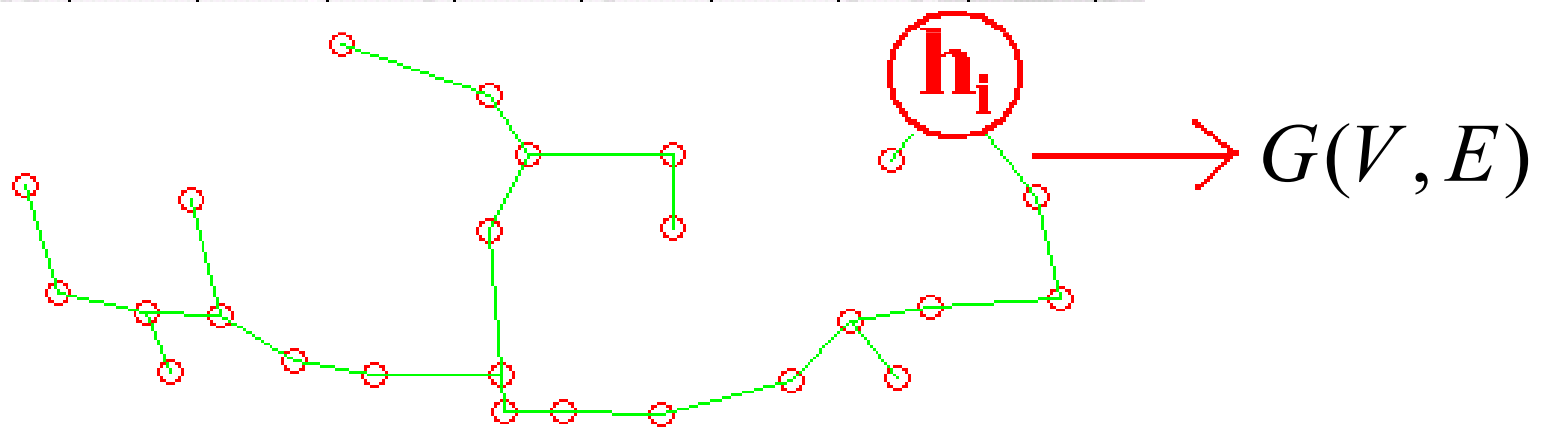
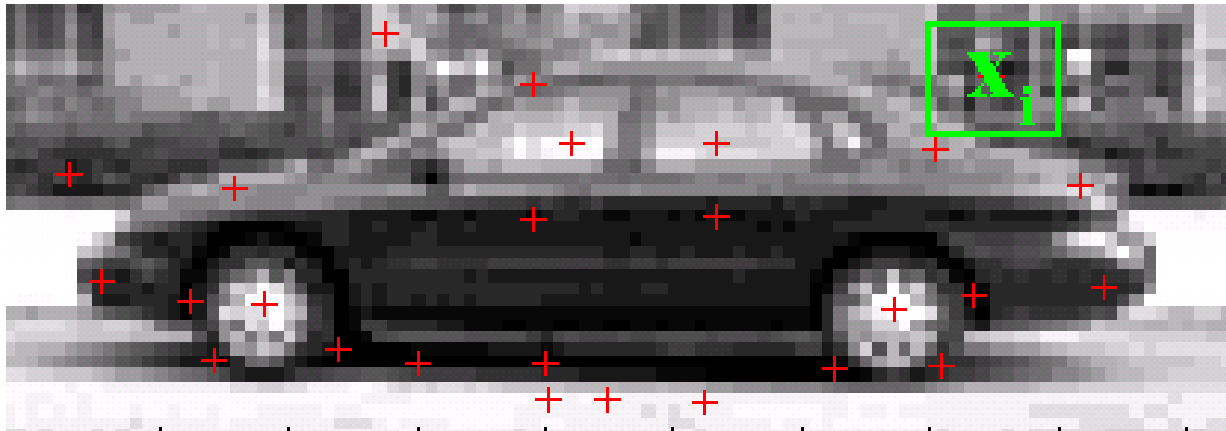
$$\phi(x_i) \in R^d$$

---

## Motivation

- Use a discriminative model.
  - Spatial dependencies between parts.
  - It is convenient to use an intermediate discrete hidden variable.
  - Potential of learning semantically-meaningful parts.
  - Framework for investigating which part structures emerge.
-

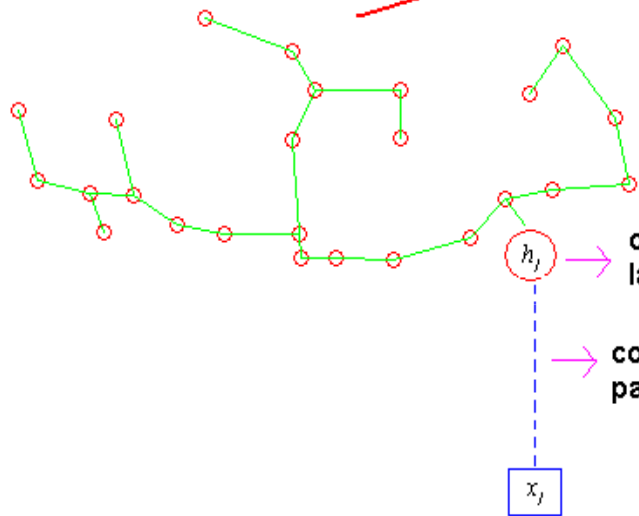
# Graph Structure



# Feature Functions

$G(V, E)$  is a minimum spanning tree.

Weight  $(i, j)$  = distance between patches  $x_i$  and  $x_j$



compatibility between a pair of part labels and a category

compatibility between a part label and a category

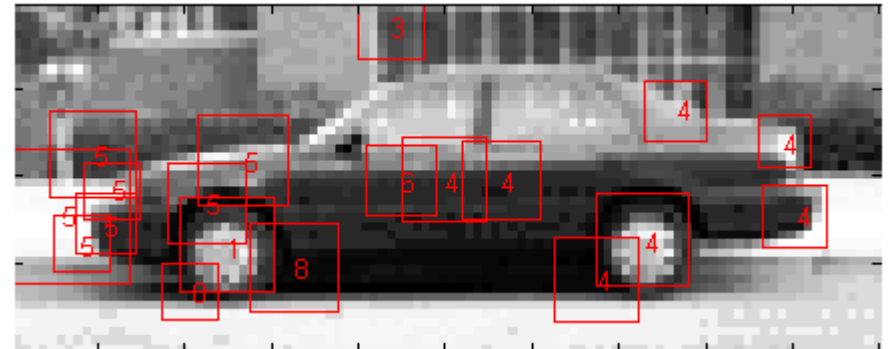
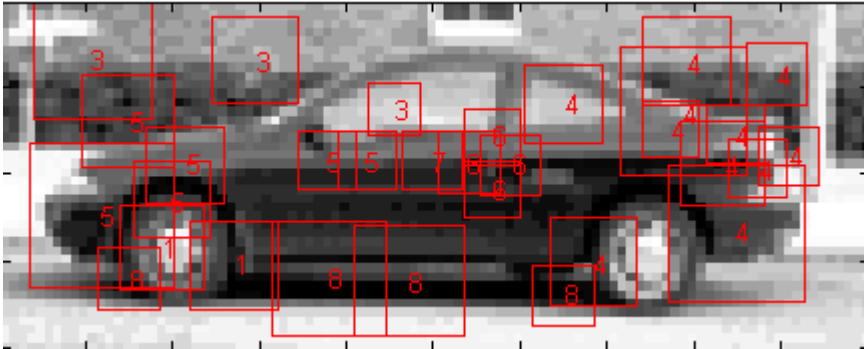
compatibility between a patch and a part label

$$f^2(j, k, y, h_j, h_k, \mathbf{x}, \theta) = \theta(y, h_j, h_k)$$

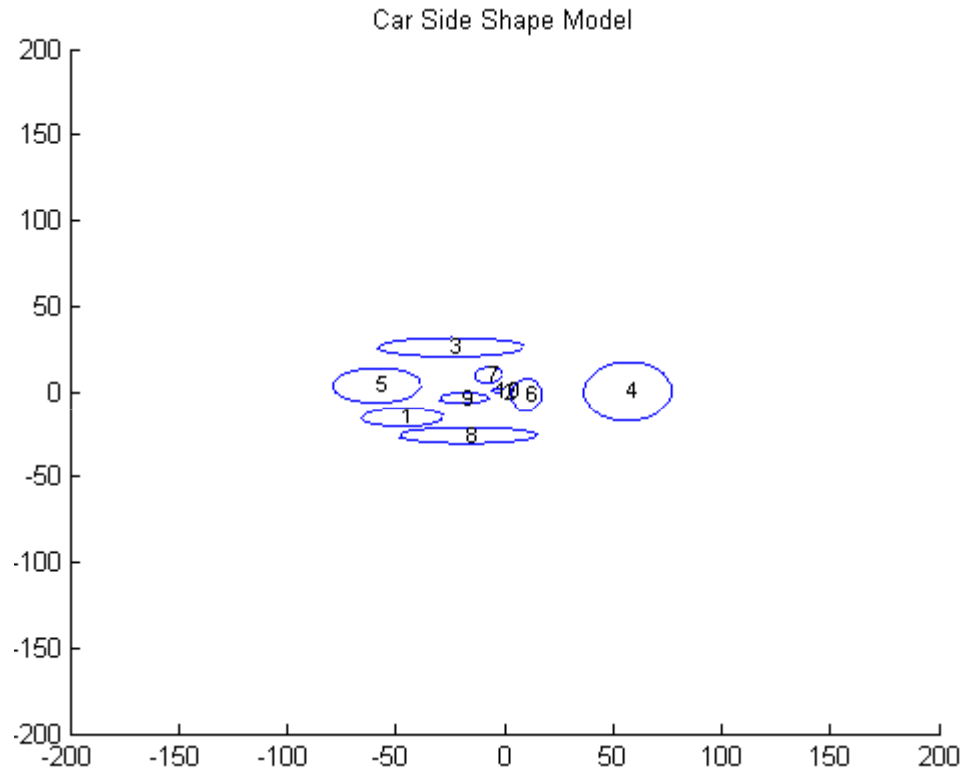
$$f^1(j, y, h_j, \mathbf{x}, \theta) = \phi(x_j) \bullet \theta(h_j) + \theta(y, h_j)$$

obtained with Lowe's detector (textured regions)  
SIFT features (describes the texture of the image region).  
Patch description also includes relative location.

# Viterbi Configuration



# Learning Shape



---

## Conclusions

- ❑ Factorized Linear Models generalize linear prediction models to the setting of structure prediction.
- ❑ In standard linear prediction, finding the argmax and computing gradients is trivial. In structure prediction it involves inference.
- ❑ Factored representations allow for efficient inference algorithms (most times based on dynamic programming)
- ❑ Conditional Random Fields are an instance of this framework

## Future Work

- ❑ Better Algorithms for training HCRFs
-