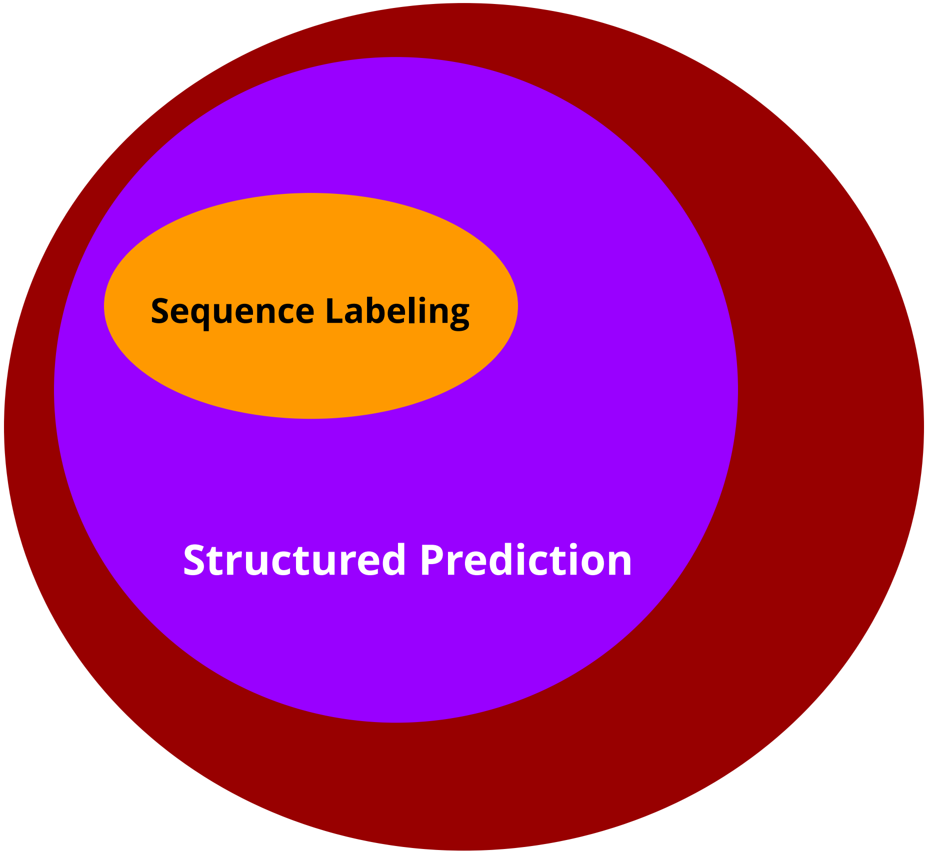
50.040
Natural Language Processing

Lu, Wei

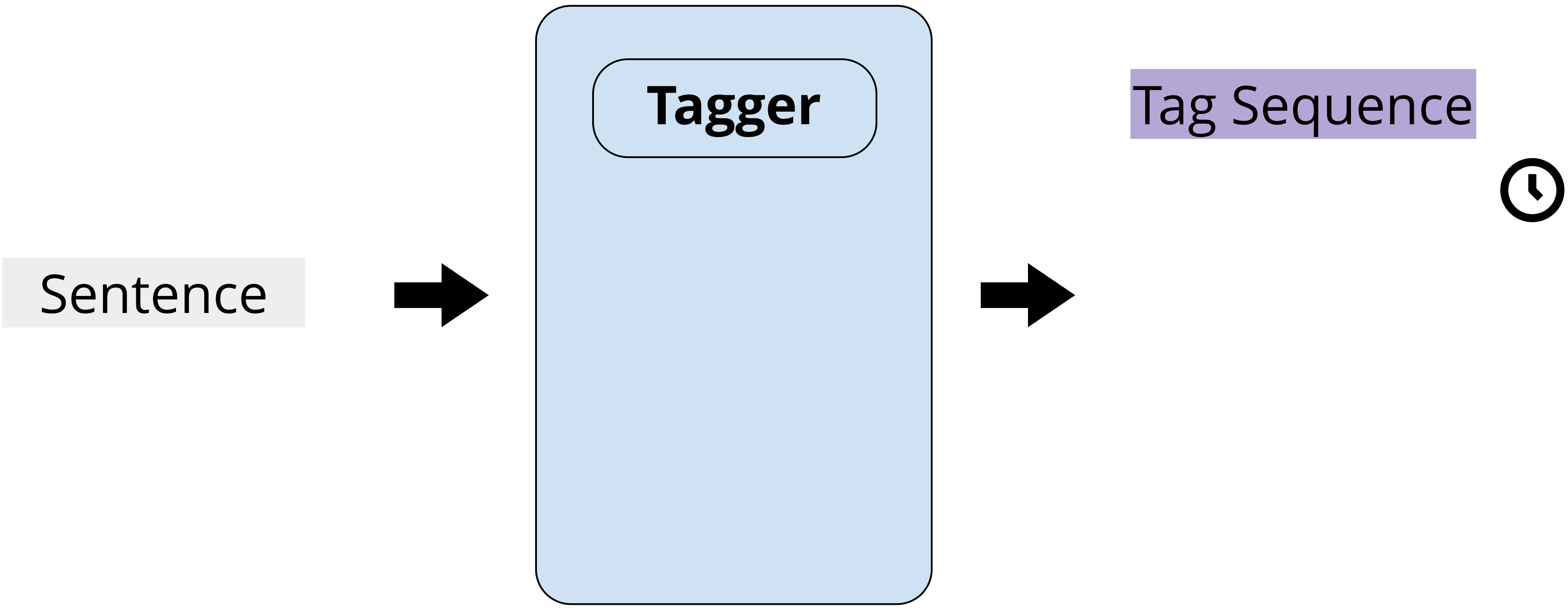
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Tasks in NLP



Supervised

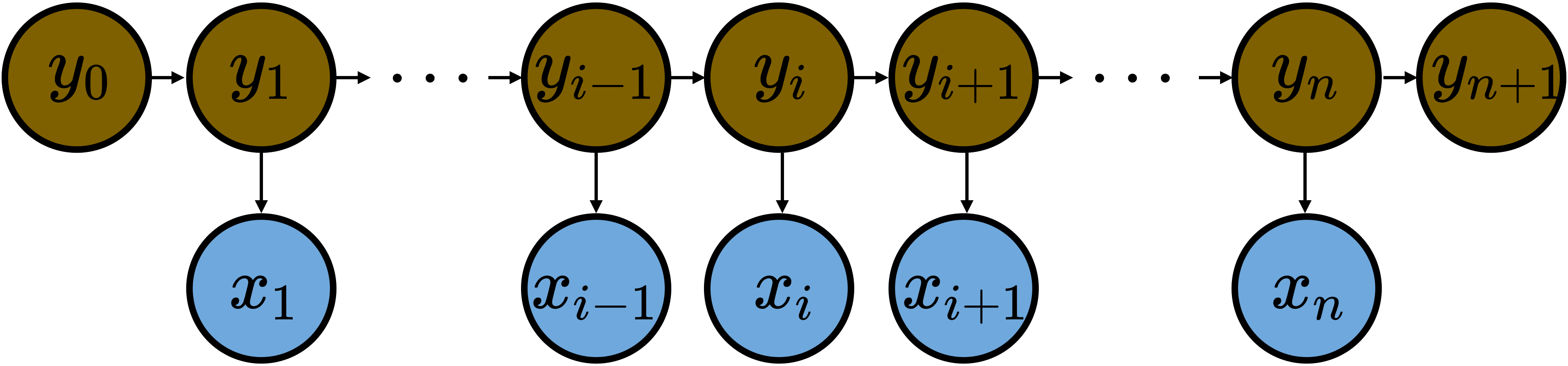
Sequence Labeling



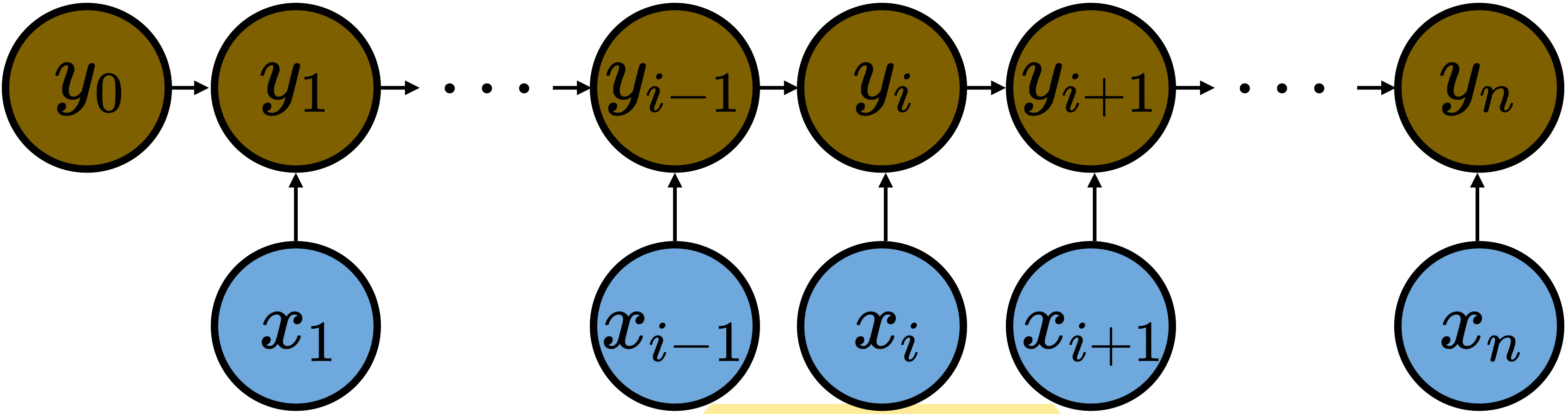
Structured Space $\mathcal{S}_i$ Structured Prediction Model M_ℓ Structured Space $\mathcal{S}_o$

$$\mathsf{M}_\ell : \mathcal{S}_i \rightarrow \mathcal{S}_o$$

HMM vs MEMM



HMM

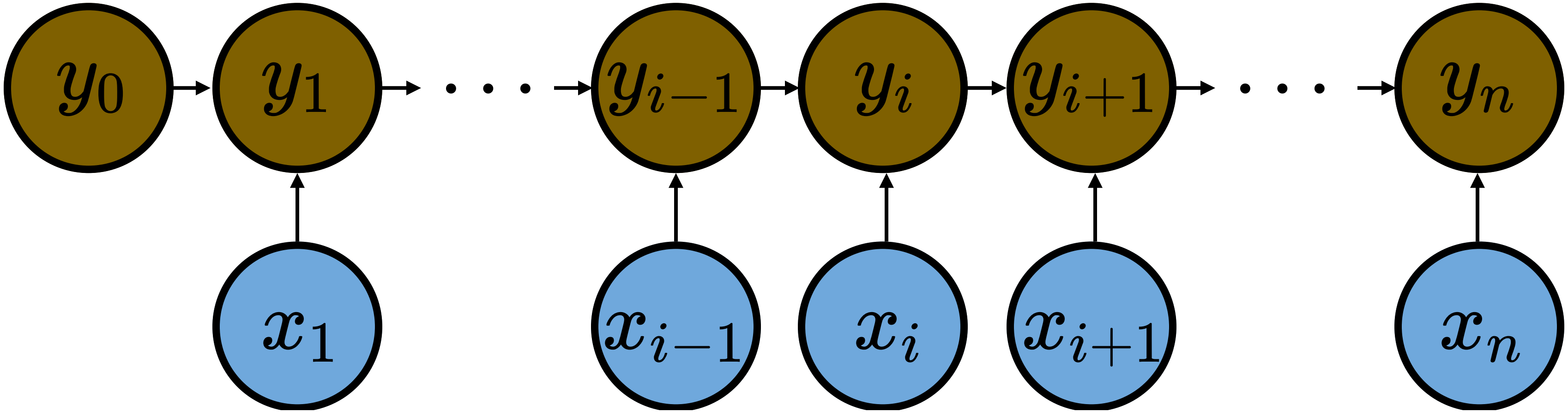


MEMM

Maximum Entropy Markov Model (McCallum et al., 2000)

$$p(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} p(y_i \mid x_i, y_{i-1})$$

$$\boxed{p(y_i \mid x_i, y_{i-1}) \propto \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_i, y_{i-1}, y_i)\right)}$$



Maximum Entropy

$$p^*(\boldsymbol{y}|\boldsymbol{x}) = \max_{p(\boldsymbol{y}|\boldsymbol{x})} \mathcal{H}(p(\boldsymbol{y}|\boldsymbol{x}))$$

$$\max_{p(\boldsymbol{y}|\boldsymbol{x})} \mathcal{H}(p(\boldsymbol{y}|\boldsymbol{x}))$$

$$\text{subject to:}$$

1. It has to be a valid probability distribution:

$$\sum_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}) = 1$$

2. While we prefer a flat/uniform distribution, we shall respect the empirical features that we observed:

$$\mathbf{E}_{p(\boldsymbol{y}|\boldsymbol{x})}[f_k(\boldsymbol{x}, \boldsymbol{y})] = \mathbf{E}_{\hat{p}(\boldsymbol{y}|\boldsymbol{x})}[f_k(\boldsymbol{x}, \boldsymbol{y})]$$

The distribution we are looking for

The empirical distribution

Maximum Entropy

$$p(\boldsymbol{y} \mid \boldsymbol{x}_i) = \frac{\exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta})}{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}$$

$$\min_{p(\boldsymbol{y}|\boldsymbol{x}_i)} \max_{\boldsymbol{\theta}} -\mathcal{H}(p(\boldsymbol{y}|\boldsymbol{x}_i))$$
$$-\sum_{k=1}^{K} \lambda_k \Big(\mathbf{E}_{p(\boldsymbol{y}|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y})] - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i) \Big)$$
$$-\lambda_0 \Big(\sum_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}) - 1 \Big)$$

$$\Downarrow$$

$$\min_{\boldsymbol{\theta}} -\sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

Maximum Entropy

$$\min_{\boldsymbol{\theta}} - \sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

Consider the i-th instance

Z

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \lambda_k} = \frac{\partial \log \boxed{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}}{\partial \lambda_k} - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

Maximum Entropy

$$\min_{\boldsymbol{\theta}} - \sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

Consider the i-th instance

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \lambda_k} = \frac{\partial \log \underbrace{\boxed{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}}_{Z}}{\partial \lambda_k} - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

$$\frac{\partial \log \sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{\partial \lambda_k} = \frac{1}{Z} \cdot \frac{\partial \sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{\partial \lambda_k}$$

$$= \frac{1}{Z} \cdot \sum_{\boldsymbol{y}'} \frac{\partial \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{\partial \lambda_k}$$

$$= \sum_{\boldsymbol{y}'} \frac{\exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{Z} f_k(\boldsymbol{x}_i, \boldsymbol{y}')$$

$$= \mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)} \left[f_k(\boldsymbol{x}_i, \boldsymbol{y}') \right]$$

Maximum Entropy

$$\min_{\boldsymbol{\theta}} - \sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \lambda_k} = \frac{\partial \log \sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{\partial \lambda_k} - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

$$= \ldots$$

$$= \mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)} [f_k(\boldsymbol{x}_i, \boldsymbol{y}')] - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

Setting this to zero is exactly one of the two constraints in the optimization problem!

Maximum Entropy

$$\min_{\boldsymbol{\theta}} - \sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \lambda_k} = \frac{\partial \log \sum_{\boldsymbol{y'}} \exp(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}') \cdot \boldsymbol{\theta})}{\partial \lambda_k} - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

$$= \ldots$$

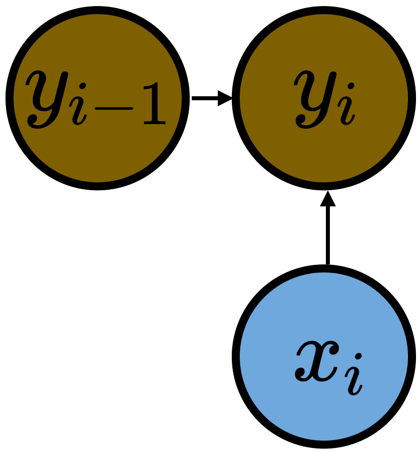
$$= \mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)} [f_k(\boldsymbol{x}_i, \boldsymbol{y}')] - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

We can then use (batch) stochastic gradient descent to optimize the objective!

Maximum Entropy Markov Model

$$p(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} p(y_i \mid x_i, y_{i-1})$$

Each term is a locally trained Maximum Entropy classifier!



Maximum Entropy Markov Model

One Theoretical Limitation

$$p(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} p(y_i \mid x_i, y_{i-1})$$

The model predicts one tag at a time (i.e., performs local normalization). Since we are modeling sequences, is it possible to predict a complete sequence at a time (i.e., perform global normalization)?

$$p(y_1, \ldots \boldsymbol{y} \ldots, y_n \mid x_1, \ldots \boldsymbol{x} \ldots, x_n)$$

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto \exp(\mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) \cdot \boldsymbol{\theta})$$

?

Conditional Random Fields (Lafferty et al., 2001)

Let's now try to model complete sequences

This is now a complete word sequence

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{\theta})}{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y}')\cdot\boldsymbol{\theta})}$$

This is now a complete tag sequence

Can we calculate the denominator efficiently?

Learning in CRF

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{\theta})}{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y}')\cdot\boldsymbol{\theta})}$$

$$\min_{\boldsymbol{\theta}} -\sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \lambda_k} = \mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')] - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

Hmm... there are exponentially many possible $\boldsymbol{y}'$ here, we shall find a way to calculate this term efficiently!

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Let's consider a very simple type of feature

$$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th tag is “N” and } (j+1)\text{-th tag is “V”}]\!]$$

$$\boxed{= 1}$$

$\boldsymbol{y}'$	**N**	**A**	**N**	**V**	**V**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Let's consider a very simple type of feature

$$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}') =$$

$$\sum_j [\![j\text{-th tag is "N" and } (j+1)\text{-th tag is "V"}]\!]$$

$$\boxed{= 2}$$

$\boldsymbol{y}'$	**N**	**V**	**N**	**V**	**V**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

						$p(\boldsymbol{y}'\|\boldsymbol{x}_i)$	$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}')$
$\boldsymbol{y}'$	**A**	**A**	**A**	**A**	**A**	0.001	0
			$\cdots$				
	A	**N**	**V**	**N**	**V**	0.007	2
	A	**N**	**V**	**D**	**N**	0.002	1
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*		

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

						$p(\boldsymbol{y}'\|\boldsymbol{x}_i)$	$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}')$
$\boldsymbol{y}'$	**A**	**A**	**A**	**A**	**A**	0.001	0
			$\cdots$				
	A	**N**	**V**	**N**	**V**	0.007	2
	A	**N**	**V**	**D**	**N**	0.002	1
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*		

However, it is not feasible to enumerate all the $\boldsymbol{y}'$ sequences. Why?

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

						$p(\boldsymbol{y}'\|\boldsymbol{x}_i)$	$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}')$
$\boldsymbol{y}'$	**A**	**A**	**A**	**A**	**A**	0.001	0
			$\cdots$				
	A	**N**	**V**	**N**	**V**	0.007	2
	A	**N**	**V**	**D**	**N**	0.002	1
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*		

Hold on a second... isn't this the same as what we did in HMM for calculating the expected counts for transitions?

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

						$p(\boldsymbol{y}'\|\boldsymbol{x}_i)$	$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}')$
$\boldsymbol{y}'$	**A**	**A**	**A**	**A**	**A**	0.001	0
			$\cdots$				
	A	**N**	**V**	**N**	**V**	0.007	2
	A	**N**	**V**	**D**	**N**	0.002	1
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*		

Hold on a second... isn't this the same as what w
HMM for calculating the expected co sitions?

Forward-backward algorithm!

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Another type of simple features

$$f_{102}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th word is "a" and } j\text{-th tag is "D"}]\!]$$

$$\boxed{= 1}$$

$\boldsymbol{y}'$	**A**	**N**	**V**	**D**	**N**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Another type of simple features

$$f_{102}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th word is "a" and } j\text{-th tag is "D"}]\!]$$

$$\boxed{= 0}$$

$\boldsymbol{y}'$	**A**	**A**	**A**	**A**	**A**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)} [f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Yet another feature

$$f_{1782}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th word is "flies" and } j\text{-th tag is "N" and } (j+1)\text{-th tag is "V"}]\!]$$

$$\boxed{= 1}$$

$\boldsymbol{y}'$	A	N	V	D	N
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

Yet another feature

$$f_{1782}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\textstyle\sum_j [\![j\text{-th word is "flies" and } j\text{-th tag is "N"}$$
$$\text{and } (j+1)\text{-th tag is "V"}]\!]$$

$$\boxed{= 0}$$

$\boldsymbol{y}'$	**N**	**A**	**N**	**V**	**V**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

Features for CRF

$$\mathbf{E}_{p(\boldsymbol{y}'|\boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')]$$

$$f_{102}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th word is “a” and } j\text{-th tag is “D”}]\!]$$

$$f_{918}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th tag is “N” and } (j+1)\text{-th tag is “V”}]\!]$$

$$f_{1782}(\boldsymbol{x}_i, \boldsymbol{y}') =$$
$$\sum_j [\![j\text{-th word is “flies” and } j\text{-th tag is “N” and } (j+1)\text{-th tag is “V”}]\!]$$

$\boldsymbol{y}'$	**N**	**A**	**N**	**V**	**V**
$\boldsymbol{x}_i$	*Fruit*	*flies*	*like*	*a*	*banana*

We shall define *local* features so as to apply efficient dynamic programming algorithms for calculating expected feature counts!

Learning in CRF

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y})\cdot\boldsymbol{\theta})}{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y}')\cdot\boldsymbol{\theta})}$$

$$\min_{\boldsymbol{\theta}} - \sum_i \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$$

$$\frac{\partial - \log p(\boldsymbol{y}_i | \boldsymbol{x}_i)}{\partial \theta_k} = \mathbf{E}_{p(\boldsymbol{y}' | \boldsymbol{x}_i)}[f_k(\boldsymbol{x}_i, \boldsymbol{y}')] - f_k(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

Forward-backward algorithm

Decoding in CRF

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{\theta})}{\sum_{\boldsymbol{y}'} \exp(\mathbf{f}(\boldsymbol{x},\boldsymbol{y}') \cdot \boldsymbol{\theta})}$$

$$\begin{aligned} \boldsymbol{y}^* &= \arg\max_{\boldsymbol{y}} \log p(\boldsymbol{y} \mid \boldsymbol{x}^{(new)}) \\ &= \arg\max_{\boldsymbol{y}} \mathbf{f}(\boldsymbol{x}^{(new)}, \boldsymbol{y}) \cdot \boldsymbol{\theta} \end{aligned}$$

A new input sentence

Learned model parameters

Viterbi algorithm!

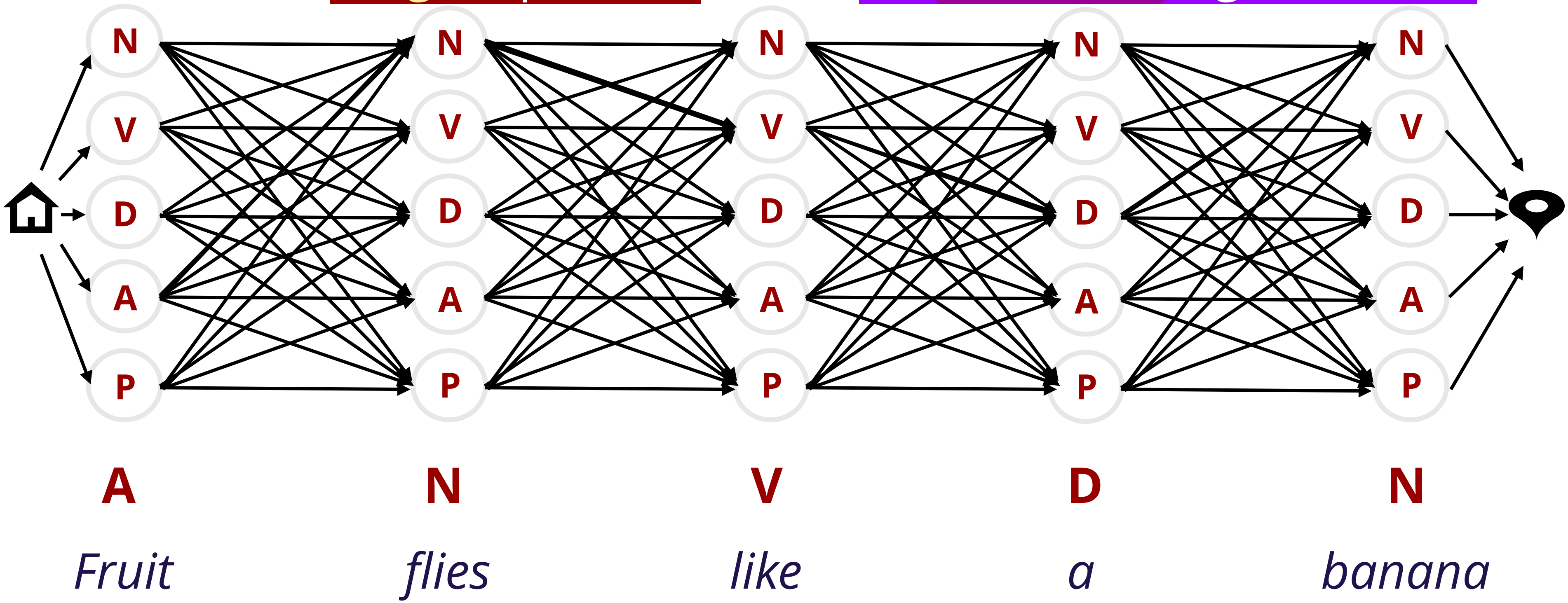
CRF

Objective Function

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right) \right)$$

Score of the gold path

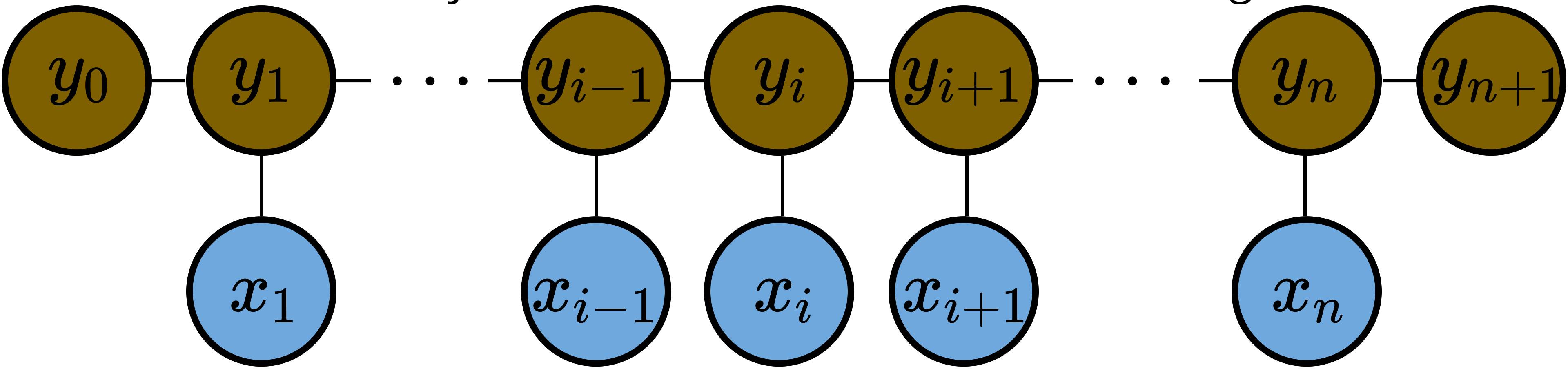
Calculated with forward-backward algorithm



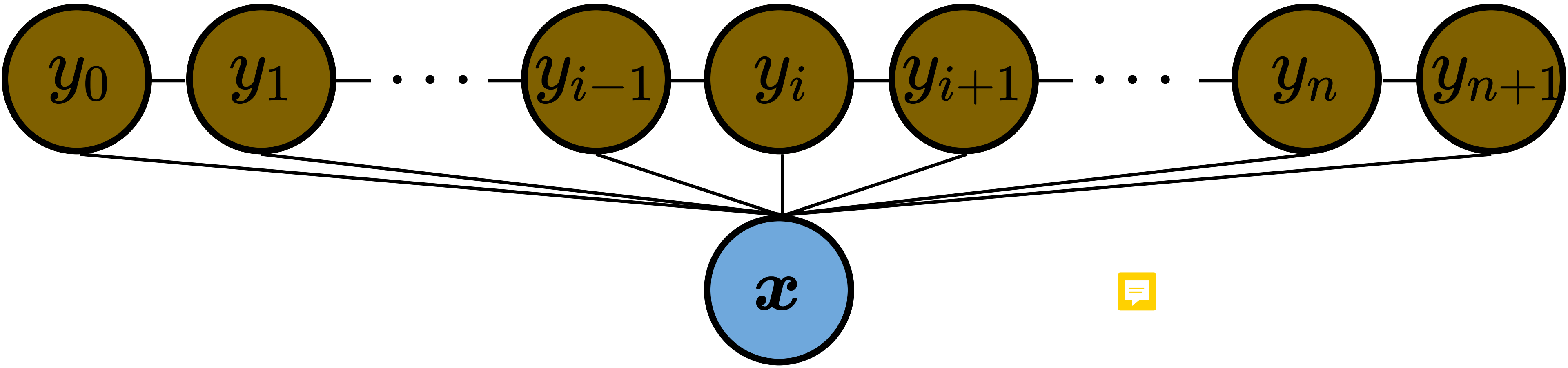
CRF

Optional

It is OK for you to think of the model as something like:



Although it actually looks something like this:



Discriminative Models
Probabilistic vs Non-Probabilistic

Probabilistic	**Non-Probabilistic**
Logistic regression	Perceptron
Softmax regression	Support vector machines

Is it possible to train a discriminative model for sequence labeling with a non-probabilisitc model such as Perceptron?

Perceptron

$\hat{y} \in \{-1, +1\}$

$$\hat{y} \leftarrow \text{sign}(\boldsymbol{x}_i \cdot \boldsymbol{\theta} + \theta_0)$$

$$\text{if } \hat{y} \neq y_i \text{ then}$$

$$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y_i \boldsymbol{x}_i$$

$$\quad \theta_0 \leftarrow \theta_0 + y_i$$

Invented more than 50 years ago.

Still popular today due to its simplicity!

Perceptron

Let's understand it a little better...

$$\hat{y} \leftarrow \arg\max_y y(\boldsymbol{x}_i \cdot \boldsymbol{\theta} + \theta_0)$$

$$\boxed{\hat{y} \leftarrow \text{sign}(\boldsymbol{x}_i \cdot \boldsymbol{\theta} + \theta_0)}$$

if $\hat{y} \neq y_i$ then

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y_i \boldsymbol{x}_i$$

$$\theta_0 \leftarrow \theta_0 + y_i$$

Perceptron

$$\hat{y} \leftarrow \arg\max_y y(\boldsymbol{x}_i \cdot \boldsymbol{\theta} + \theta_0)$$

$$\text{if } \hat{y} \neq y_i \text{ then}$$

$$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y_i \boldsymbol{x}_i$$

$$\quad \theta_0 \leftarrow \theta_0 + y_i$$

Perceptron

$$\hat{y} \leftarrow \arg\max_y \begin{pmatrix} y\boldsymbol{x}_i/2 \\ y/2 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix}$$

$$\boxed{\hat{y} \leftarrow \arg\max_y y(\boldsymbol{x}_i \cdot \boldsymbol{\theta} + \theta_0)}$$

if $\hat{y} \neq y_i$ then

$$\boxed{\begin{aligned} \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} + y_i\boldsymbol{x}_i \\ \theta_0 &\leftarrow \theta_0 + y_i \end{aligned}}$$

$$\begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix} + \begin{pmatrix} y_i\boldsymbol{x}_i \\ y_i \end{pmatrix}$$

Perceptron

$$\hat{y} \leftarrow \arg\max_y \begin{pmatrix} y\boldsymbol{x}_i/2 \\ y/2 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix}$$

if $\hat{y} \neq y_i$ then

$$\begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix} + \begin{pmatrix} y_i\boldsymbol{x}_i \\ y_i \end{pmatrix}$$

Perceptron

$$\mathbf{f}(\boldsymbol{x}_i, y) \qquad \boldsymbol{\theta}'$$

$$\hat{y} \leftarrow \arg\max_y \boxed{\begin{pmatrix} y\boldsymbol{x}_i/2 \\ y/2 \end{pmatrix}} \cdot \boxed{\begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix}}$$

$$\text{if } \hat{y} \neq y_i \text{ then}$$

$$\boxed{\begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix}} \leftarrow \boxed{\begin{pmatrix} \boldsymbol{\theta} \\ \theta_0 \end{pmatrix}} + \boxed{\begin{pmatrix} y_i\boldsymbol{x}_i \\ y_i \end{pmatrix}}$$

$$\boldsymbol{\theta}' \qquad \boldsymbol{\theta}' \qquad \mathbf{f}(\boldsymbol{x}_i, y_i) - \mathbf{f}(\boldsymbol{x}_i, \hat{y})$$

Perceptron

$$\hat{y} \leftarrow \arg\max_y \mathbf{f}(\boldsymbol{x}_i, y) \cdot \boldsymbol{\theta}'$$

$$\text{if } \hat{y} \neq y_i \text{ then}$$

$$\quad \boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}' + \mathbf{f}(\boldsymbol{x}_i, y_i) - \mathbf{f}(\boldsymbol{x}_i, \hat{y})$$

This allows us to make a generalization for the variable y : from a binary output to a structured output

Structured Perceptron
(Collins, 1999)

This is a structure now!

The space of structures given $\boldsymbol{x}_i$

$$\hat{\boldsymbol{y}}_i \leftarrow \arg\max_{\boldsymbol{y} \in \mathbf{GEN}(\boldsymbol{x}_i)} \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}$$

$$\text{if } \hat{\boldsymbol{y}}_i \neq \boldsymbol{y}_i \text{ then}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \mathbf{f}(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)$$

Structured Perceptron

$$\hat{\boldsymbol{y}}_i \leftarrow \arg\max_{\boldsymbol{y} \in \mathbf{GEN}(\boldsymbol{x}_i)} \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}$$

$$\text{if } \hat{\boldsymbol{y}}_i \neq \boldsymbol{y}_i \text{ then}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \mathbf{f}(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)$$

We shall make sure searching in the space defined by the $\mathbf{GEN}(\boldsymbol{x}_i)$ function is efficient.

Viterbi algorithm!

Define similar local features as what we did in CRF!

Structured Perceptron

$$\hat{\boldsymbol{y}}_i \leftarrow \arg\max_{\boldsymbol{y} \in \mathbf{GEN}(\boldsymbol{x}_i)} \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}$$

$$\text{if } \hat{\boldsymbol{y}}_i \neq \boldsymbol{y}_i \text{ then}$$
$$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \mathbf{f}(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)$$

This update rule is essentially stochastic gradient descent with learning rate 1!

What is the underlying objective function?

$$-\,\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} \;+\; \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right)$$

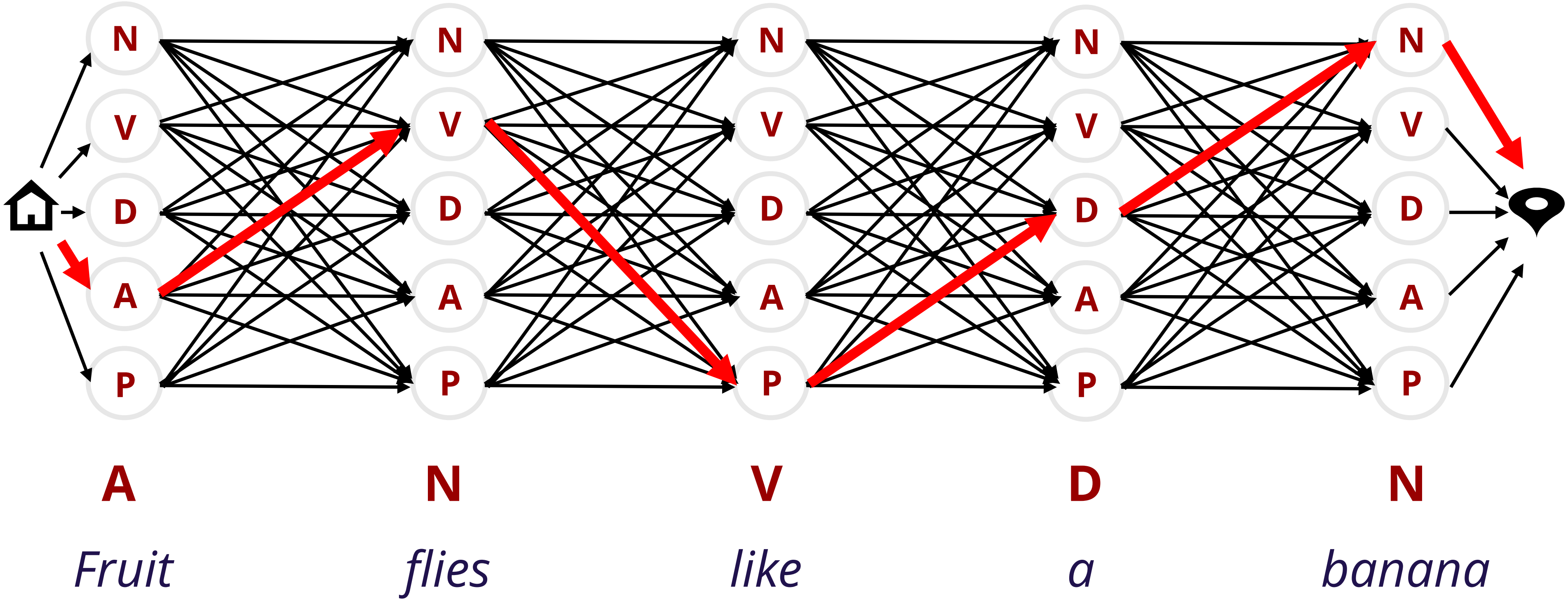
Structured Perceptron

Objective Function

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\, \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} \;+\; \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$

Score of the gold path

Score of the best path

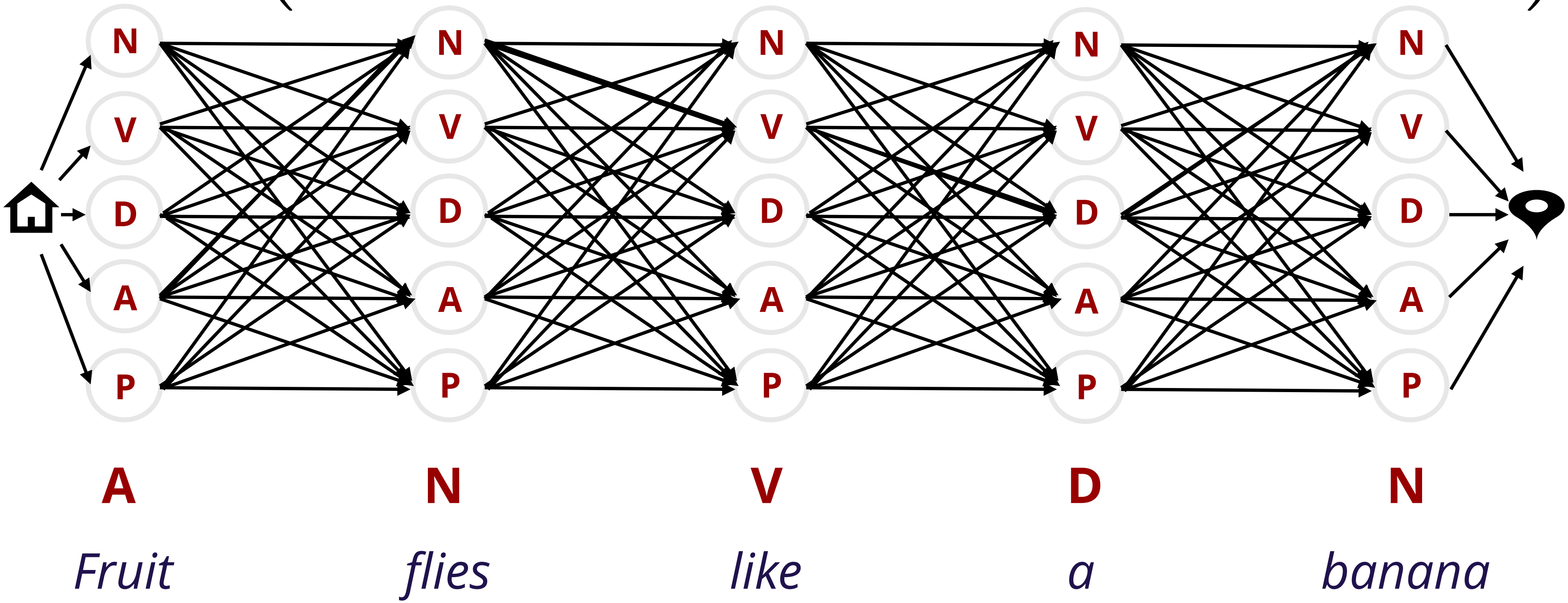


CRF vs SP

Comparison

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$



CRF vs SP

Comparison

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \boxed{\log \sum_{\boldsymbol{y}} \exp} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \boxed{\max_{\boldsymbol{y}}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$

The difference!

CRF vs SP

Comparison

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right) \right)$$

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right) \right)$$

$$\mathcal{Y} = \{1, 3, 4, 8, 18\}$$

$$\log \sum_{y \in \mathcal{Y}} \exp(y) = ?$$

$$\max_{y \in \mathcal{Y}} (y) = ?$$

CRF vs SP

Comparison

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right) \right)$$

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right) \right)$$

$$\mathcal{Y} = \{1, 3, 4, 8, 18\}$$

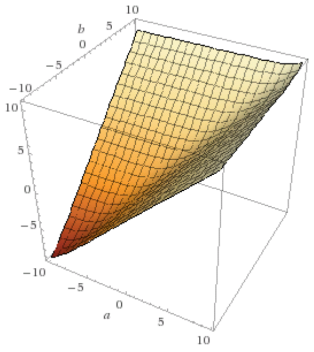
$$\log \sum_{y \in \mathcal{Y}} \exp(y) = 18.000465777..$$

$$\max_{y \in \mathcal{Y}} (y) = 18$$

CRF vs SP
Comparison

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right)\right)$$

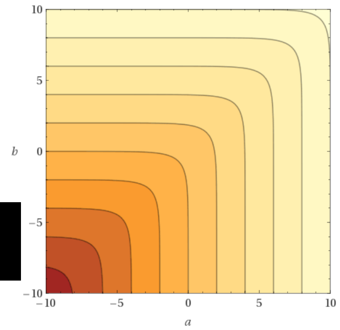
$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \max_{\boldsymbol{y}} \left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta}\right)\right)$$

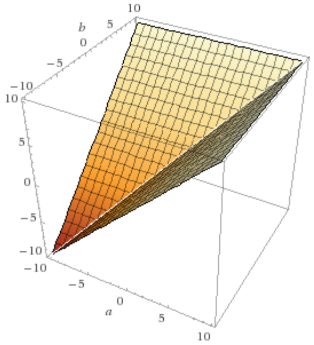


Computed by Wolfram|Alpha

$$\log(\exp(a) + \exp(b))$$

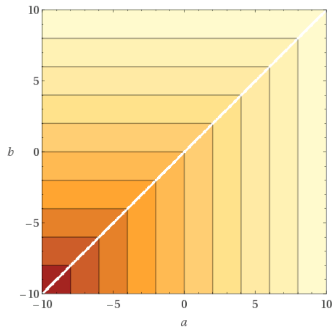
A "soft"/"smooth" version of max!





Computed by Wolfram|Alpha

$$\max(a, b)$$

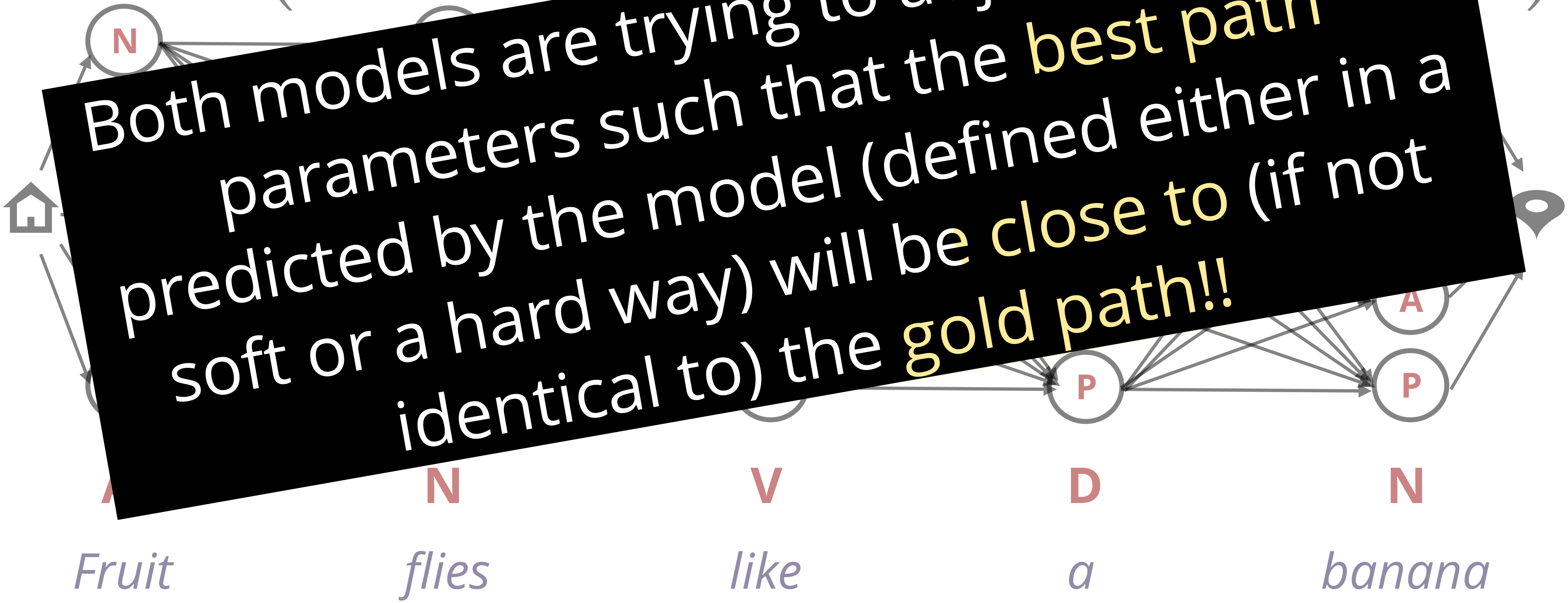


CRF vs SP
Comparison

$$\min_{\theta} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$

$$\min_{\theta} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \quad \boldsymbol{\theta}) \right)$$

Both models are trying to adjust the model parameters such that the best path predicted by the model (defined either in a soft or a hard way) will be close to (if not identical to) the gold path!!



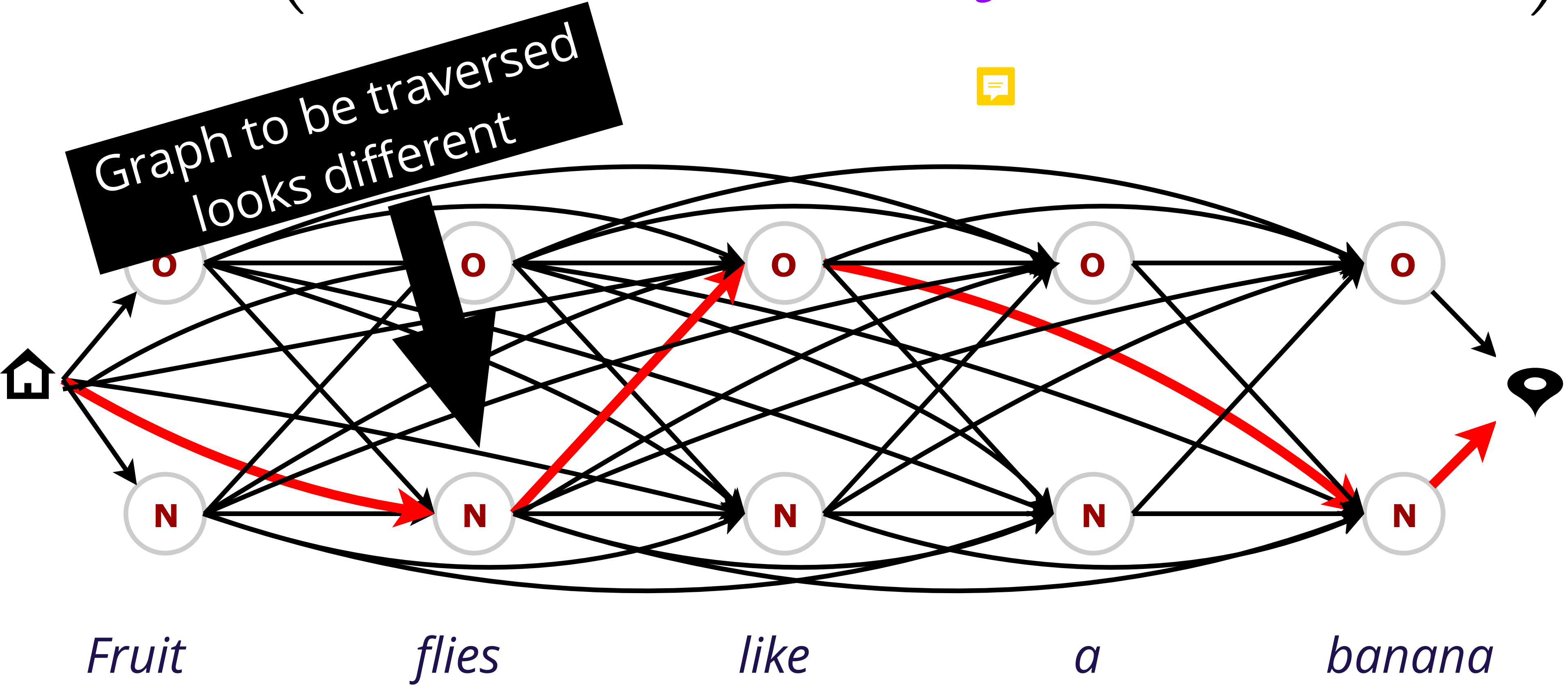
Sequence Labeling

Model	Pros	Cons
Hidden Markov Model (HMM)	Probabilistic, efficient training	Unable to exploit linguistic features
Maximum Entropy Markov Model (MEMM)	Probabilistic, able to exploit features	Training is slower than HMM, local normalization
Conditional Random Fields (CRF)	Probabilistic, able to exploit features, global normalization	Training may be slower than MEMM
Structured Perceptron (SP)	Able to exploit features, performance comparable to CRF	Non-probabilistic

Optional

Semi-Markov CRF (Sarawagi & Cohen, 2005)

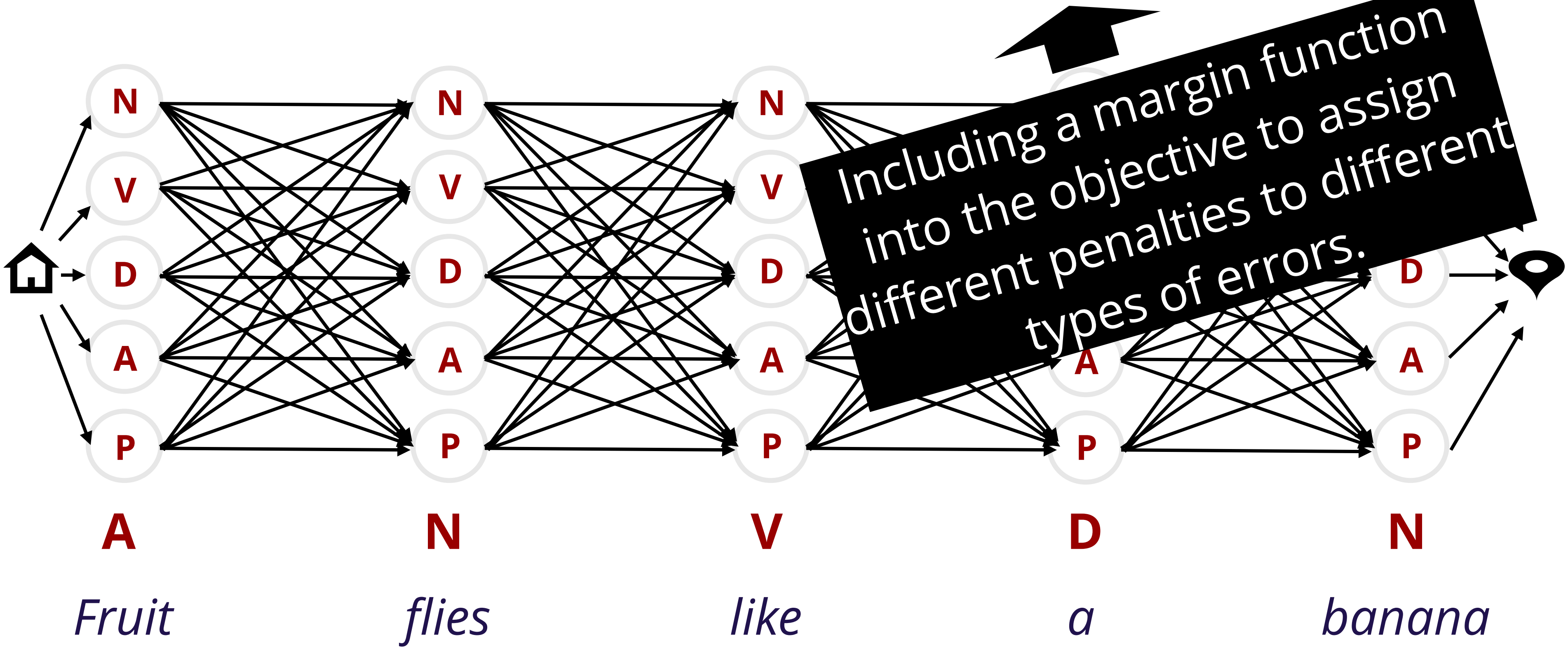
$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \log \sum_{\boldsymbol{y}} \exp\left(\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$



Optional

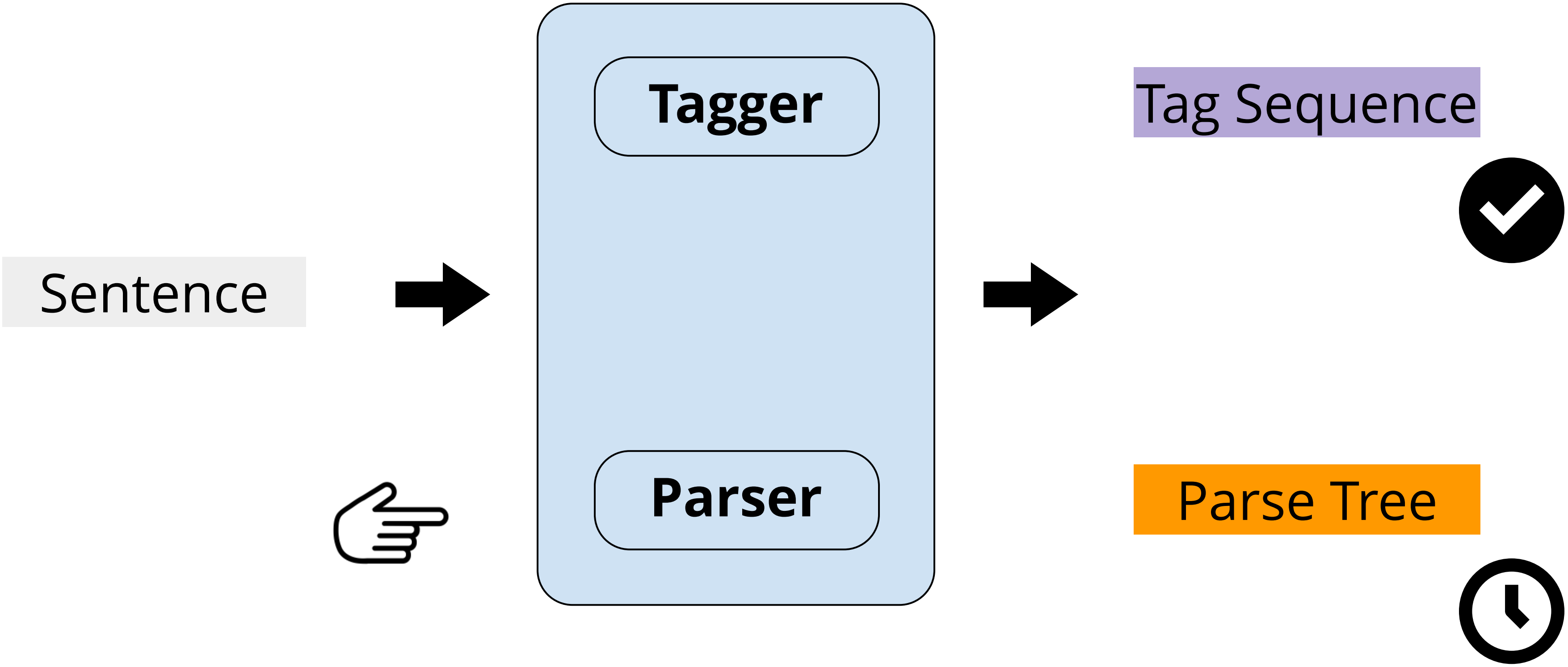
Structural SVM (Tsochantaridis et al., 2005)

$$\min_{\boldsymbol{\theta}} \sum_i \left(-\mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \boldsymbol{\theta} + \max_{\boldsymbol{y}} \left(\Delta(\boldsymbol{y}_i, \boldsymbol{y}) + \mathbf{f}(\boldsymbol{x}_i, \boldsymbol{y}) \cdot \boldsymbol{\theta} \right) \right)$$



Including a margin function into the objective to assign different penalties to different types of errors.

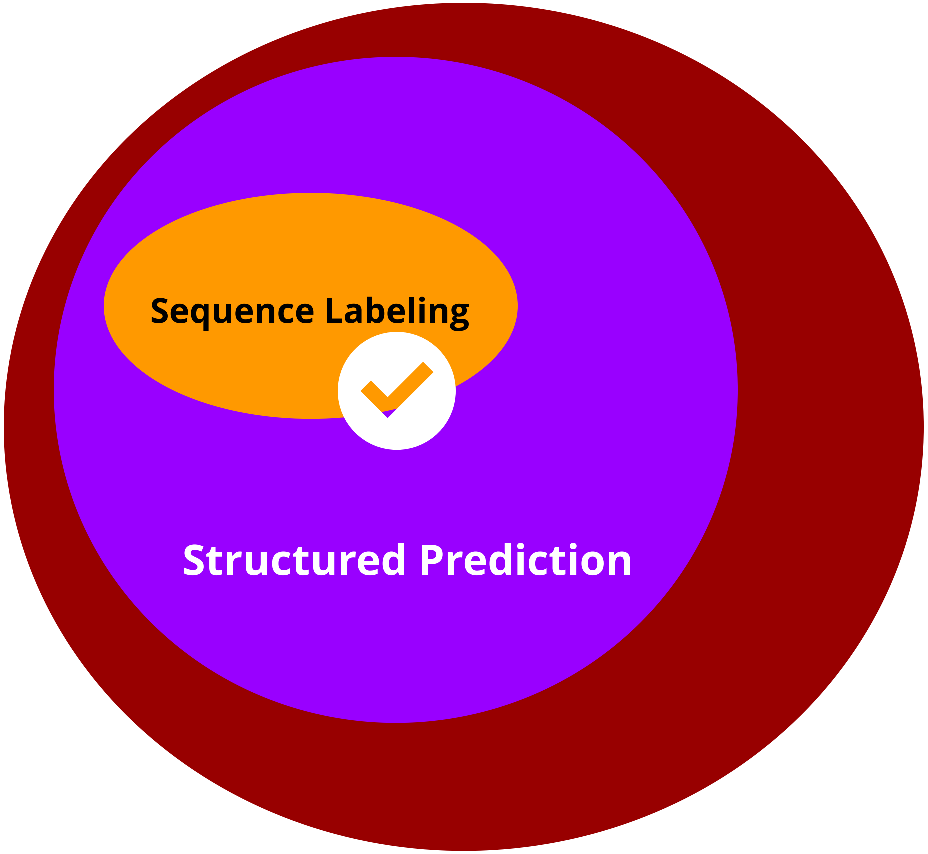
Structured Prediction



Structured Space $\mathcal{S}_i$ Structured Prediction Model M Structured Space $\mathcal{S}_o$

$$\mathrm{M} : \mathcal{S}_i \rightarrow \mathcal{S}_o$$

Tasks in NLP



Supervised