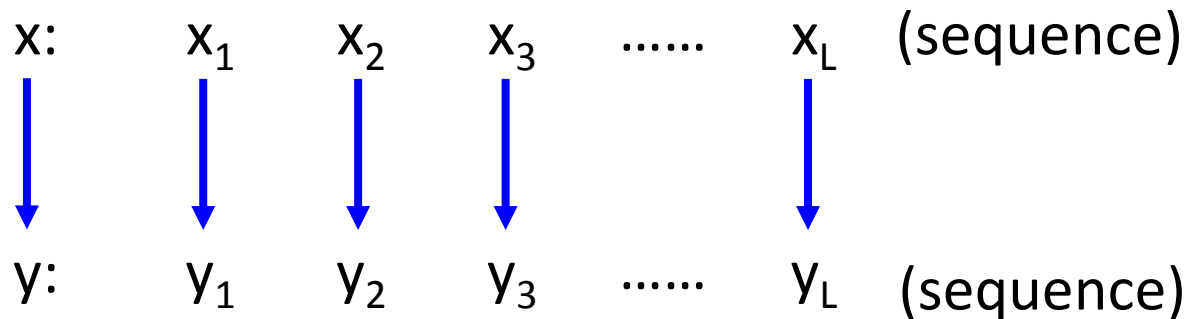


Sequence Labeling Problem

Sequence Labeling

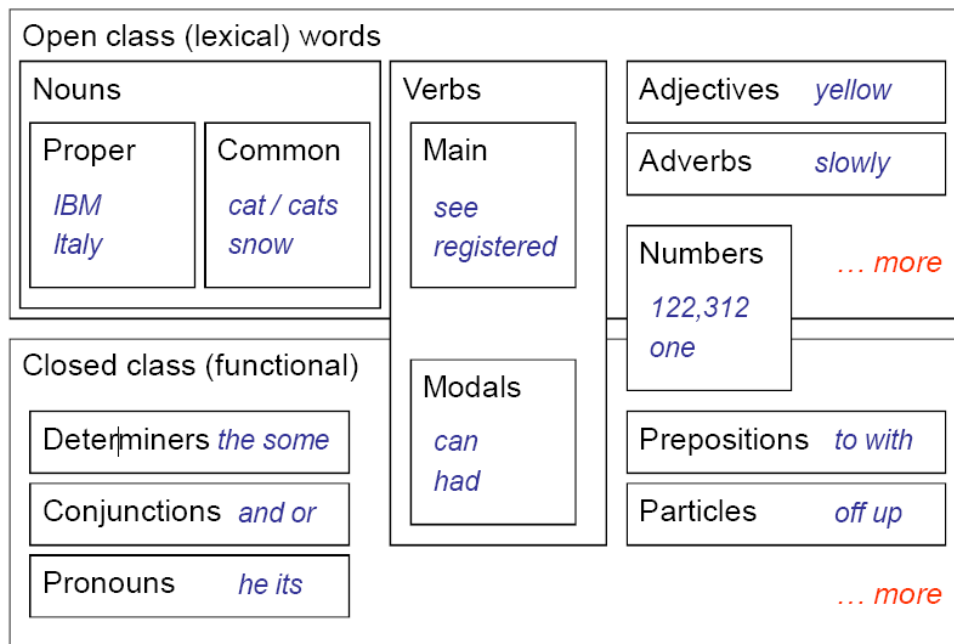
$$f : \underset{\text{Sequence}}{X} \rightarrow \underset{\text{Sequence}}{Y}$$



RNN can handle this task, but there are other methods based on structured learning (two steps, three problems).

Example Task

- POS tagging
 - Annotate each word in a sentence with a part-of-speech.



John saw the saw.

↓ ↓ ↓ ↓

PN V D N

- Useful for subsequent syntactic parsing and word sense disambiguation, etc.

Example Task

- POS tagging

John saw the saw.
↓ ↓ ↓ ↓
PN V D N

The problem cannot be solved
without considering the sequences.

- “saw” is more likely to be a verb V rather than a noun N
- However, the second “saw” is a noun N because a noun N is more likely to follow a determiner.

Outline

Hidden Markov Model (HMM)



Conditional Random Field (CRF)



Structured Perceptron/SVM



Towards Deep Learning

Outline

Hidden Markov Model (HMM)



Conditional Random Field (CRF)



Structured Perceptron/SVM



Towards Deep Learning

HMM

- How you generate a sentence?

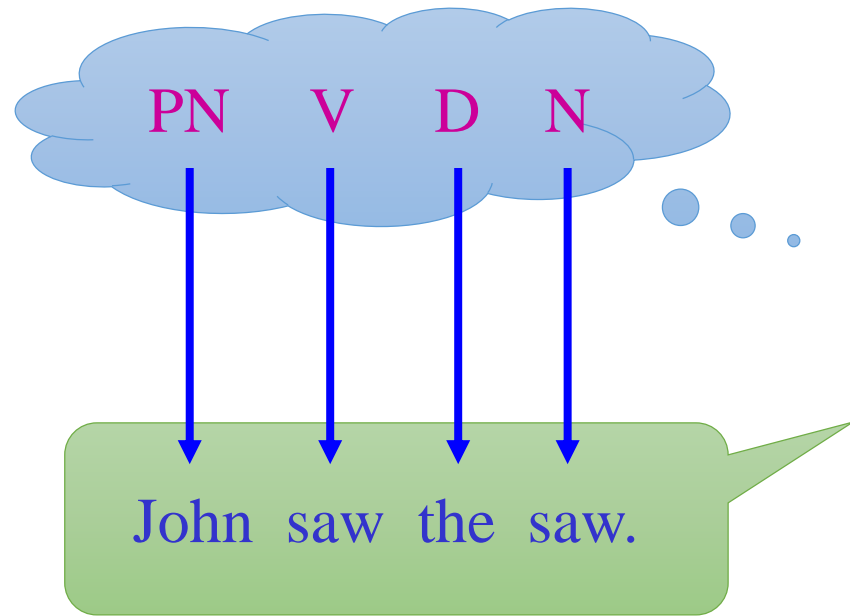
Just the assumption
of HMM

Step 1

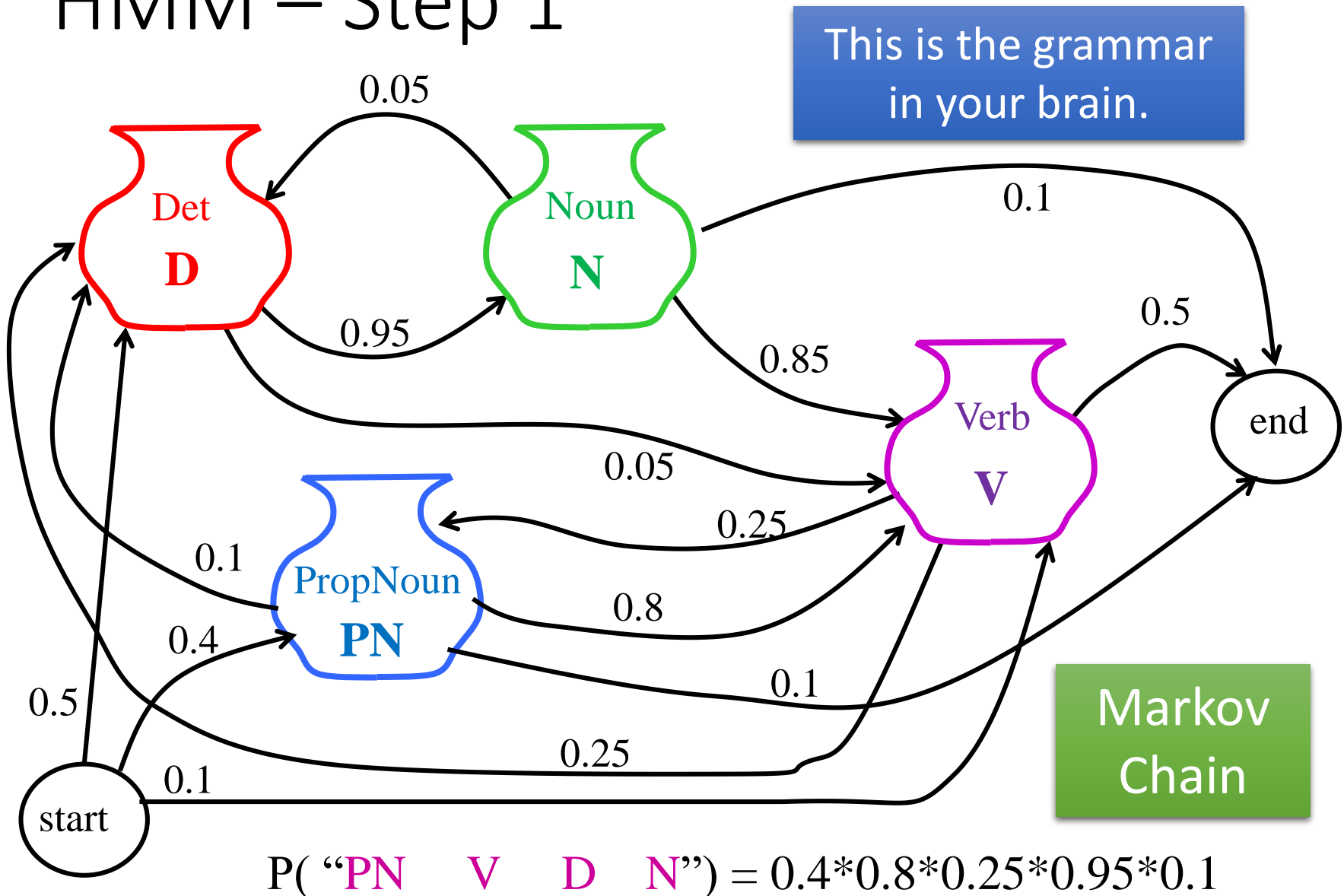
- Generate a POS sequence
- Based on the grammar

Step 2

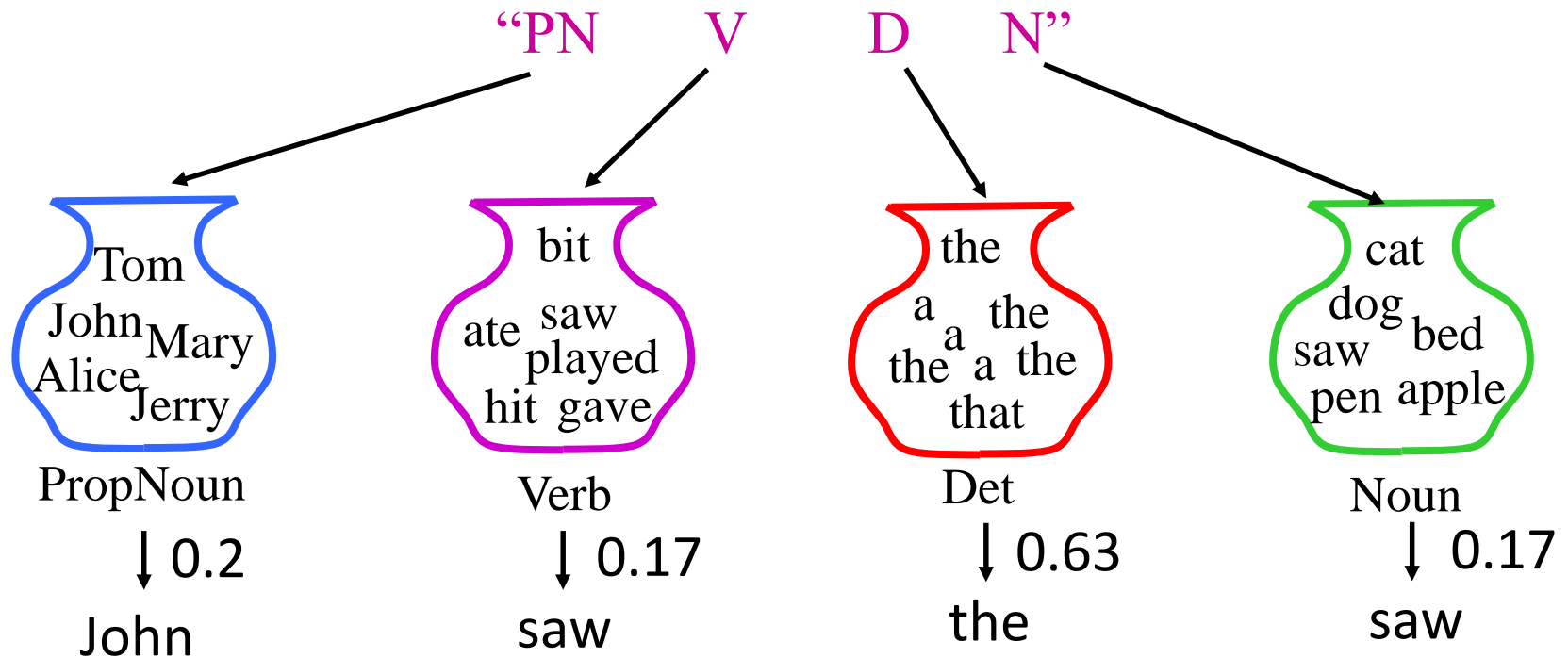
- Generate a sentence based on the POS sequence
- Based on a dictionary



HMM – Step 1

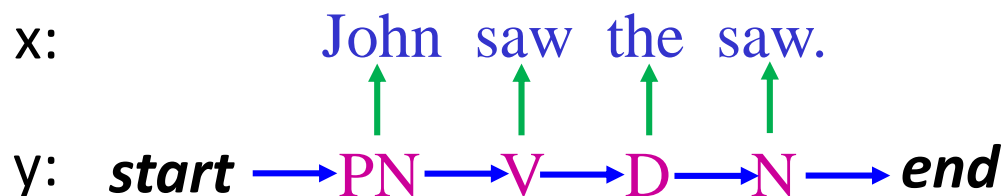


HMM – Step 2



$$P(\text{"John saw the saw"} \mid \text{"PN V D N"}) \\ = 0.2 * 0.17 * 0.63 * 0.17$$

HMM



$$P(x,y) = P(y)P(x|y)$$

$$\begin{aligned} P(y) &= P(PN|start) \\ &\times P(V|PN) \\ &\times P(D|V) \\ &\times P(N|D) \end{aligned}$$

$$\begin{aligned} P(x|y) &= P(John|PN) \\ &\times P(saw|V) \\ &\times P(the|D) \\ &\times P(saw|N) \end{aligned}$$

HMM

x: John saw the saw.

$$x = x_1, x_2 \cdots x_L$$

y: PN V D N

$$y = y_1, y_2 \cdots y_L$$

$$P(x, y) = P(y)P(x|y)$$

Step 1

$$P(y) = P(y_1 | start) \times \prod_{l=1}^{L-1} P(y_{l+1} | y_l) \times P(end | y_L)$$

Transition probability

Step 2

$$P(x|y) = \prod_{l=1}^L P(x_l | y_l)$$

Emission probability

HMM

– Estimating the probabilities

- How can I know $P(V|PN)$, $P(\text{saw}|V)$?
- Obtaining from training data

Training Data:

- (x^1, \hat{y}^1) 1 Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.
- (x^2, \hat{y}^2) 2 Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.
- (x^3, \hat{y}^3) 3 Rudolph/NNP Agnew/NNP ,/, 55/CD years/NNS old/JJ and/CC chairman/NN of/IN Consolidated/NNP Gold/NNP Fields/NNP PLC/NNP ,/, was/VBD named/VBN a/DT nonexecutive/JJ director/NN of/IN this/DT British/JJ industrial/JJ conglomerate/NN ./.

⋮

HMM

– Estimating the probabilities

$$P(x, y) = \underbrace{P(y_1 | start)} \prod_{l=1}^{L-1} \underbrace{P(y_{l+1} | y_l)} \underbrace{P(end | y_L)} \prod_{l=1}^L \underbrace{P(x_l | y_l)}$$

$$\underbrace{P(y_{l+1} = s' | y_l = s)} = \frac{count(s \rightarrow s')}{count(s)}$$

(s and s' are tags)

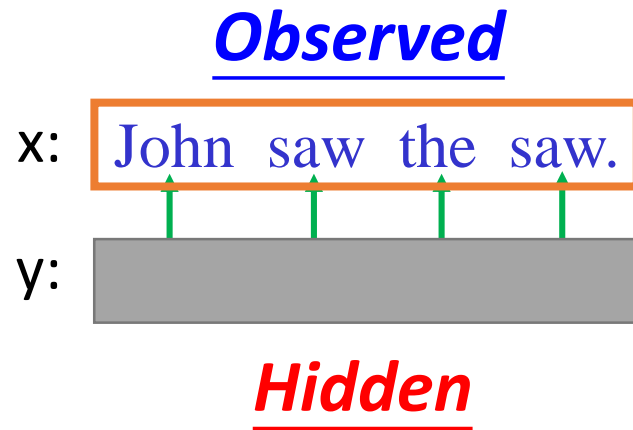
$$\underbrace{P(x_l = t | y_l = s)} = \frac{count(s \rightarrow t)}{count(s)}$$

(s is tag, and t is word)

So simple 😊

HMM – How to do POS Tagging?

- We can compute $P(x, y)$



Task: given x , find y

$$y = \arg \max_{y \in Y} P(y|x)$$

$$= \arg \max_{y \in Y} \frac{P(x, y)}{P(x)}$$

$$= \arg \max_{y \in Y} P(x, y)$$

HMM – Viterbi Algorithm

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

- Enumerate all possible y
 - Assume there are $|S|$ tags, and the length of sequence y is L
 - There are $|S|^L$ possible y
- ***Viterbi algorithm***
 - Solve the above problem with complexity $O(L|S|^2)$

HMM - Summary

Problem 1:
Evaluation



Problem 2:
Inference



Problem 3:
Training

$$F(x, y) = P(x, y) = P(y)P(x|y)$$

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

$P(y)$ and $P(x|y)$ can be simply
obtained from training data

HMM - Drawbacks

- Inference:

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

- To obtain correct results ...

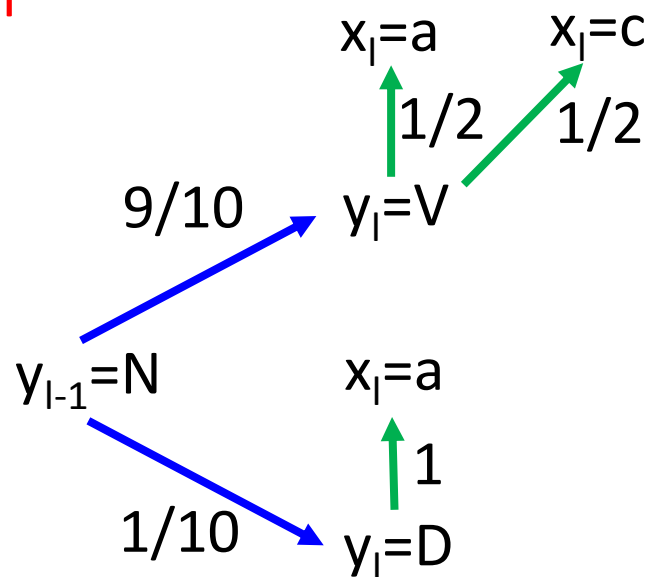
$(x, \hat{y}): P(x, \hat{y}) > \underline{P(x, y)}$ Can HMM guarantee that?
not necessarily small

Transition probability:

$P(V|N)=9/10$ $P(D|N)=1/10$

Emission probability:

$P(a|V)=1/2$ $P(a|D)=1$



HMM - Drawbacks

- Inference:

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

- To obtain correct results ...

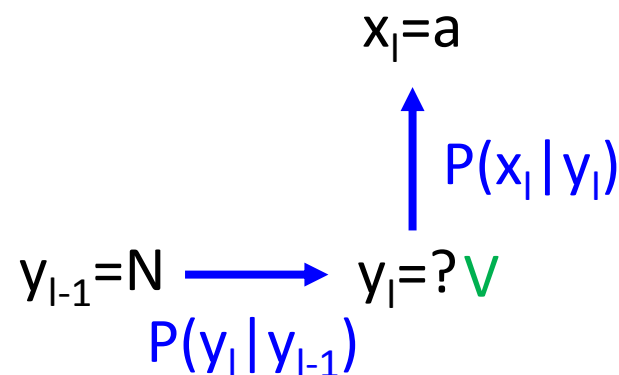
$(x, \hat{y}): P(x, \hat{y}) > \underline{P(x, y)}$ Can HMM guarantee that?
not necessarily small

Transition probability:

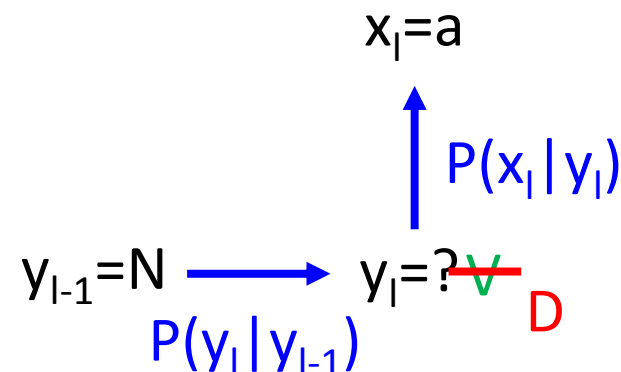
$$P(V|N)=9/10 \quad P(D|N)=1/10 \quad \dots\dots$$

Emission probability:

$$P(a|V)=1/2 \quad P(a|D)=1 \quad \dots\dots$$



HMM - Drawbacks



- Inference:

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

- To obtain correct results ...

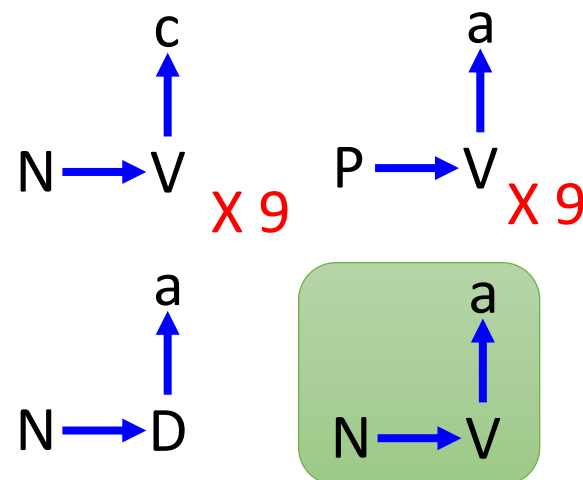
$(x, \hat{y}): P(x, \hat{y}) > \underline{P(x, y)}$ Can HMM guarantee that?
 not necessarily small

Transition probability:

$$P(V|N)=9/10 \quad P(D|N)=1/10 \quad \dots\dots$$

Emission probability:

$$P(a|V)=1/2 \quad P(a|D)=1 \quad \dots\dots$$

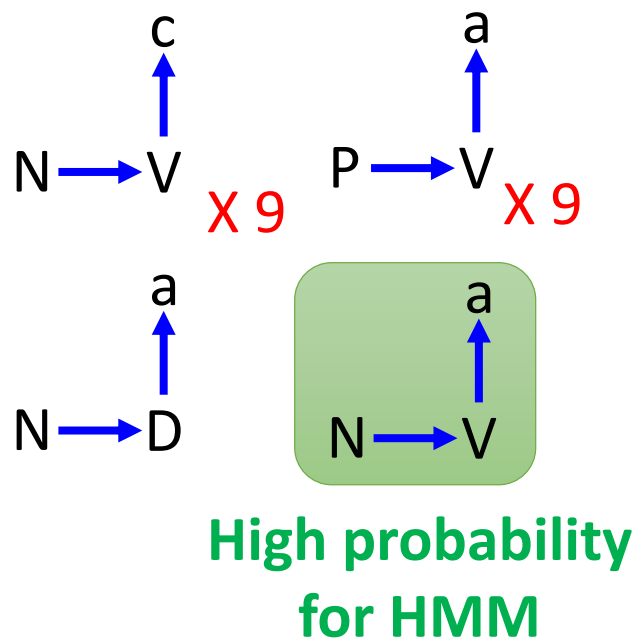


High probability
for HMM

HMM - Drawbacks

- The (x,y) never seen in the training data can have large probability $P(x,y)$.
- Benefit:
 - When there is only little training data

- More complex model can deal with this problem
- However, CRF can deal with this problem based on the same model



Outline

Hidden Markov Model (HMM)



Conditional Random Field (CRF)



Structured Perceptron/SVM



Towards Deep Learning

CRF

$$P(x, y) \propto \exp(\mathbf{w} \cdot \phi(x, y))$$

➤ $\phi(x, y)$ is a feature vector. What does it look like?

➤ \mathbf{w} is a weight vector to be learned from training data

➤ $\exp(\mathbf{w} \cdot \phi(x, y))$ is always positive, can be larger than 1


$$\begin{aligned} P(y|x) &= \frac{P(x, y)}{\sum_{y'} P(x, y')} & P(x, y) &= \frac{\exp(\mathbf{w} \cdot \phi(x, y))}{R} \\ &= \frac{\exp(\mathbf{w} \cdot \phi(x, y))}{\sum_{y' \in \mathbb{Y}} \exp(\mathbf{w} \cdot \phi(x, y'))} & &= \frac{\exp(\mathbf{w} \cdot \phi(x, y))}{Z(x)} \end{aligned}$$

$P(x, y)$ for CRF

$$P(x, y) \propto \exp(\mathbf{w} \cdot \boldsymbol{\phi}(x, y))$$

very different from HMM?

In HMM:


$$P(x, y) = P(y_1 | start) \prod_{l=1}^{L-1} P(y_{l+1} | y_l) P(end | y_L) \prod_{l=1}^L P(x_l | y_l)$$
$$\log P(x, y)$$

$$= \log P(y_1 | start) + \sum_{l=1}^{L-1} \log P(y_{l+1} | y_l) + \log P(end | y_L)$$
$$+ \sum_{l=1}^L \log P(x_l | y_l)$$

$P(x, y)$ for CRF

$$\log P(x, y) = \log P(y_1 | \text{start}) + \sum_{l=1}^{L-1} \log P(y_{l+1} | y_l) + \log P(\text{end} | y_L) + \sum_{l=1}^L \log P(x_l | y_l)$$

Log probability of
word t given tag s

Number of tag s and word t
appears together in (x, y)

$$\sum_{l=1}^L \log P(x_l | y_l) = \sum_{s, t} \log P(t | s) \times N_{s, t}(x, y)$$

Enumerate all possible tags s
and all possible word t

$P(x,y)$ for CRF

x: The dog ate the homework.
 ↓ ↓ ↓ ↓ ↓
 y: D N V D N

$$N_{D,the}(x,y) = 2$$

$$N_{N,dog}(x,y) = 1$$

$$N_{V,ate}(x,y) = 1$$

$$N_{N,homework}(x,y) = 1$$

$$N_{s,t}(x,y) = 0$$

(for any other s and t)

$$\sum_{l=1}^L \log P(x_l | y_l)$$

$$= \log P(\text{the} | D) + \log P(\text{dog} | N) + \log P(\text{ate} | V) \\ + \log P(\text{the} | D) + \log P(\text{homework} | N)$$

$$= \log P(\text{the} | D) \times 2 + \log P(\text{dog} | N) \times 1 + \log P(\text{ate} | V) \times 1 \\ + \log P(\text{homework} | N) \times 1$$

$$= \sum_{s,t} \log P(t | s) \times N_{s,t}(x,y)$$

$P(x, y)$ for CRF

$$\log P(x, y)$$

$$= \boxed{\log P(y_1 | \text{start})} + \boxed{\sum_{l=1}^{L-1} \log P(y_{l+1} | y_l)} + \boxed{\log P(\text{end} | y_L)} \\ + \sum_{l=1}^L \log P(x_l | y_l)$$

$$\log P(y_1 | \text{start}) = \sum_s \log P(s | \text{start}) \times N_{\text{start}, s}(x, y)$$

$$\sum_{l=1}^{L-1} \log P(y_{l+1} | y_l) = \sum_{s, s'} \log P(s' | s) \times N_{s, s'}(x, y)$$

$$\log P(\text{end} | y_L) = \sum_s \log P(\text{end} | s) \times N_{s, \text{end}}(x, y)$$

P(x,y) for CRF

$$\begin{aligned}
 \log P(x, y) &= \sum_{s,t} \log P(t|s) \times N_{s,t}(x, y) \\
 &+ \sum_s \log P(s|start) \times N_{start,s}(x, y) \\
 &+ \sum_{s,s'} \log P(s'|s) \times N_{s,s'}(x, y) \\
 &+ \sum_s \log P(end|s) \times N_{s,end}(x, y)
 \end{aligned}
 = \begin{bmatrix} \vdots \\ \log P(t|s) \\ \vdots \\ \vdots \\ \log P(s|start) \\ \vdots \\ \vdots \\ \log P(s'|s) \\ \vdots \\ \vdots \\ \log P(end|s) \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ N_{s,t}(x, y) \\ \vdots \\ \vdots \\ N_{start,s}(x, y) \\ \vdots \\ \vdots \\ N_{s,s'}(x, y) \\ \vdots \\ \vdots \\ N_{s,end}(x, y) \\ \vdots \end{bmatrix}$$

$$= w \cdot \phi(x, y)$$

$$P(x, y) = \exp(w \cdot \phi(x, y))$$

P(x,y) for CRF

$$P(x,y) \propto \exp(w \cdot \phi(x,y))$$

However, we do not give w any constraints during training

$$\phi(x,y) = \begin{bmatrix} \vdots \\ N_{s,t}(x,y) \\ \vdots \\ N_{start,s}(x,y) \\ \vdots \\ N_{s,s'}(x,y) \\ \vdots \\ N_{s,end}(x,y) \\ \vdots \end{bmatrix}$$

$$w = \begin{bmatrix} \vdots \\ w_{s,t} \\ \vdots \\ w_{start,s} \\ \vdots \\ w_{s,s'} \\ \vdots \\ w_{s,end} \\ \vdots \end{bmatrix}$$

$\rightarrow \log P(x_i = t | y_i = s)$
 $P(x_i = t | y_i = s) = e^{w_{s,t}}$
means

$\rightarrow \log P(s | start)$
 $P(s | start) = e^{w_{start,s}}$
means

$\rightarrow \log P(y_i = s' | y_{i-1} = s)$
 $P(y_i = s' | y_{i-1} = s) = e^{w_{s,s'}}$
means

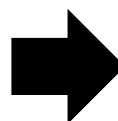
$\rightarrow \log P(end | s)$
... ..

Feature Vector

- What does $\phi(x, y)$ look like?

x: The dog ate the homework.
↓ ↓ ↓ ↓ ↓
y: D N V D N

- $\phi(x, y)$ has two parts
 - Part 1: relations between tags and words
 - Part 2: relations between tags
- If there are $|S|$ possible tags,
 $|L|$ possible words
Part 1 has $|S| \times |L|$ dimensions



Part 1	Value
D, the	2
D, dog	0
D, ate	0
D, homework	0
.....
N, the	0
N, dog	1
N, ate	0
N, homework	1
.....
V, the	0
V, dog	0
V, ate	1
V, homework	0
.....

Feature Vector

- What does $\phi(x, y)$ look like?

x: The dog ate the homework.
 ↓ ↓ ↓ ↓ ↓
 y: D N V D N

- $\phi(x, y)$ has two parts
 - Part 1: relations between tags and words
 - Part 2: relations between tags

$N_{s,s'}(x, y)$: Number of tags s and s' consecutively in (x, y)

$N_{D,D}(x, y) \rightarrow$

$N_{D,N}(x, y) \rightarrow$

Part 2	Value
D, D	0
D, N	2
D, V	0
.....
N, D	0
N, N	0
N, V	1
.....
V, D	1
V, N	0
V, V	0
.....
Start, D	1
Start, N	0
.....
End, D	0
End, N	1

Feature Vector

- What does $\phi(x, y)$ look like?

x: The dog ate the homework.
 ↓ ↓ ↓ ↓ ↓
 y: D N V D N

- $\phi(x, y)$ has two parts
 - Part 1: relations between tags and words
 - Part 2: relations between tags

If there are $|S|$ possible tags,
 $|S| \times |S| + 2 \times |S|$ dimensions

Define any $\phi(x, y)$ you like!

Part 2	Value
D, D	0
D, N	2
D, V	0
.....
N, D	0
N, N	0
N, V	1
.....
V, D	1
V, N	0
V, V	0
.....
Start, D	1
Start, N	0
.....
End, D	0
End, N	1

$$P(y|x)$$

$$= \frac{P(x, y)}{\sum_{y'} P(x, y')}$$

CRF – Training Criterion

- Given training data: $\{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \dots (x^N, \hat{y}^N)\}$
- Find the weight vector w^* maximizing objective function $O(w)$:

$$w^* = \arg \max_w O(w) \quad O(w) = \sum_{n=1}^N \log P(\hat{y}^n | x^n)$$

$$\log P(\hat{y}^n | x^n) = \log \underbrace{P(x^n, \hat{y}^n)}_{\text{Maximize what we observe}} - \log \underbrace{\sum_{y'} P(x^n, y')}_{\text{Minimize what we don't observe}}$$

Maximize what
we observe

Minimize what we
don't observe

CRF – Gradient Ascent

Gradient descent

Find a set of parameters θ minimizing cost function $C(\theta)$

$$\theta \rightarrow \theta - \eta \nabla C(\theta)$$

Opposite direction of the gradient

Gradient Ascent

Find a set of parameters θ maximizing objective function $O(\theta)$

$$\theta \rightarrow \theta + \eta \nabla O(\theta)$$

The same direction of the gradient

CRF - Training

$$O(w) = \sum_{n=1}^N \log P(\hat{y}^n | x^n) = \sum_{n=1}^N O^n(w)$$

Compute

$$\nabla O^n(w) = \begin{bmatrix} \vdots \\ \partial O^n(w) / \partial w_{s,t} \\ \vdots \\ \partial O^n(w) / \partial w_{s,s'} \\ \vdots \end{bmatrix}$$

Let me show $\frac{\partial O^n(w)}{\partial w_{s,t}}$

$\frac{\partial O^n(w)}{\partial w_{s,s'}}$ very similar

CRF - Training

$$P(y'|x^n) = \frac{\exp(w \cdot \phi(x^n, y'))}{Z(x^n)}$$

$$w_{s,t} \rightarrow w_{s,t} + \eta \frac{\partial O(w)}{\partial w_{s,t}}$$

After some math

Can be computed by Viterbi algorithm as well

$$\frac{\partial O^n(w)}{\partial w_{s,t}} = \underbrace{N_{s,t}(x^n, \hat{y}^n)}_{\text{training}} - \underbrace{\sum_{y'} P(y'|x^n) N_{s,t}(x^n, y')}_{\text{current model}}$$

If word t is labeled by tag s in training examples (x^n, \hat{y}^n) , then increase $w_{s,t}$

If word t is labeled by tag s in (x^n, y') which not in training examples, then decrease $w_{s,t}$

$$P(y'|x^n) = \frac{\exp(w \cdot \phi(x^n, y'))}{Z(x^n)}$$

CRF - Training

$$\nabla O(w) = \phi(x^n, \hat{y}^n) - \sum_{y'} P(y'|x^n) \phi(x^n, y')$$

Stochastic Gradient Ascent

Random pick a data (x^n, \hat{y}^n)

$$w \rightarrow w + \eta \left(\phi(x^n, \hat{y}^n) - \sum_{y'} P(y'|x^n) \phi(x^n, y') \right)$$

CRF – Inference

- Inference

$$y = \arg \max_{y \in Y} P(y|x) = \arg \max_{y \in Y} P(x, y)$$

$$= \arg \max_{y \in Y} w \cdot \phi(x, y) \quad \text{Done by Viterbi as well}$$

$$P(x, y) \propto \exp(w \cdot \phi(x, y))$$

CRF v.s. HMM

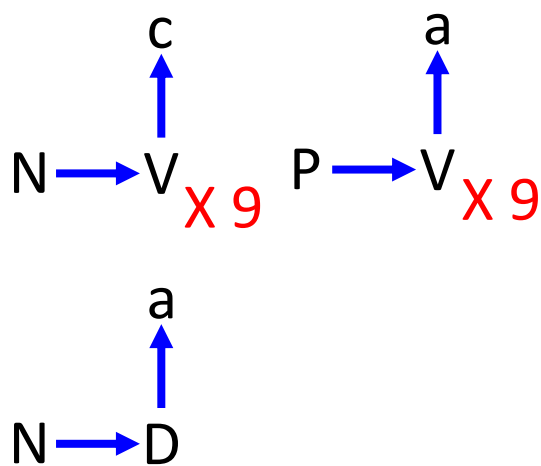
- CRF: increase $P(x, \hat{y})$, decrease $P(x, y')$

HMM does not do that

- To obtain correct results ...

$$(x, \hat{y}): P(x, \hat{y}) > P(x, y)$$

CRF more likely to achieve that than HMM



HMM:

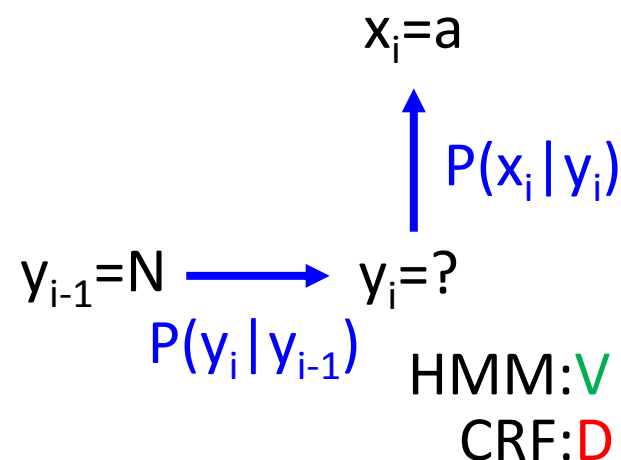
$$P(V | N) = 9/10$$

$$P(D | N) = 1/10$$

$$P(a | V) = 1/2 \rightarrow 0.1$$

$$P(a | D) = 1$$

CRF:



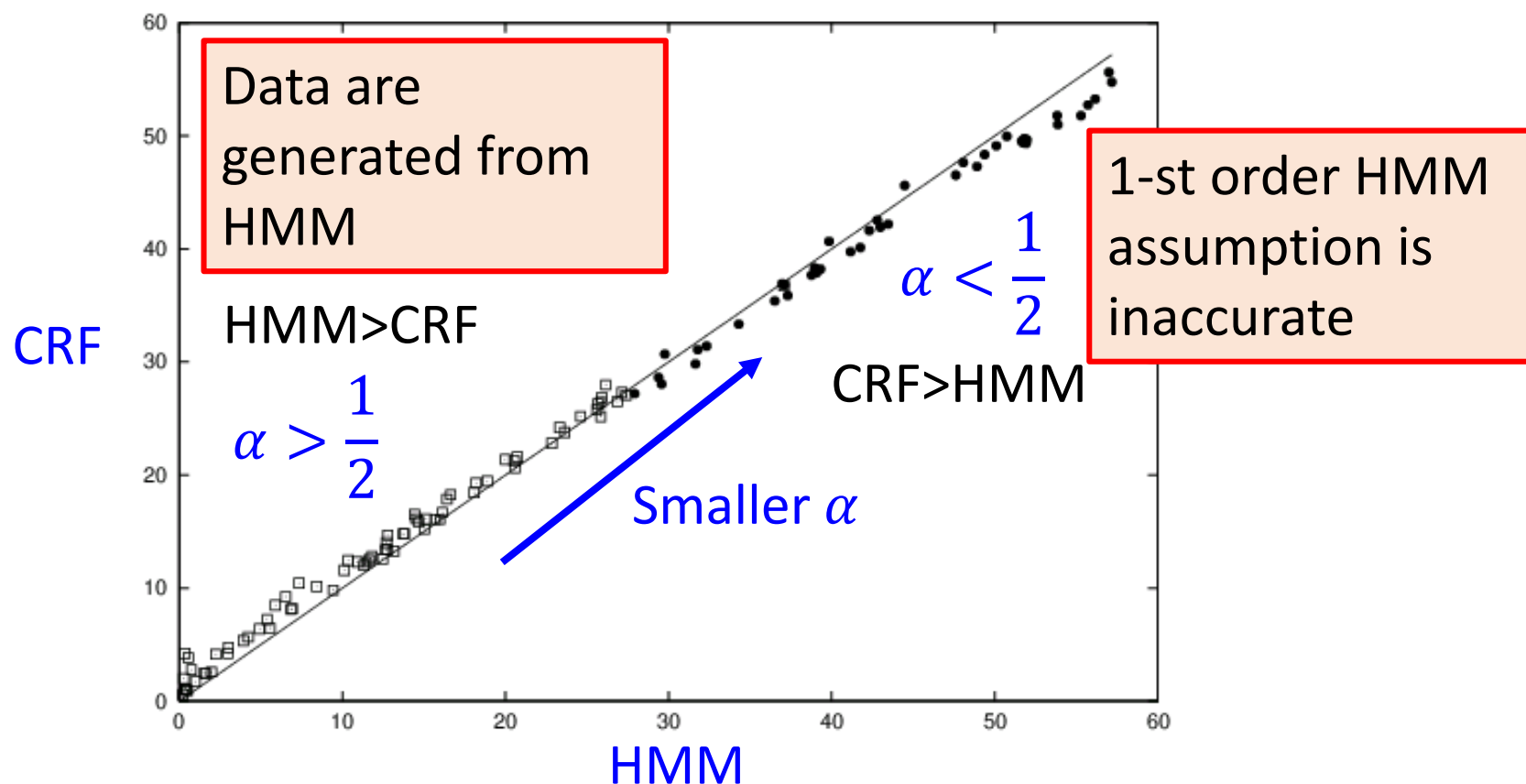
HMM: **V**
CRF: **D**

Synthetic Data

- $x_i \in \{a - z\}, y_i \in \{A - E\}$
- Generating data from a mixed-order HMM
 - Transition probability:
 - $\alpha P(y_i | y_{i-1}) + (1 - \alpha) P(y_i | y_{i-1}, y_{i-2})$
 - Emission probability:
 - $\alpha P(x_i | y_i) + (1 - \alpha) P(x_i | y_i, x_{i-1})$
- Comparing HMM and CRF
 - All the approaches only consider 1-st order information
 - Only considering the relation of y_{i-1} and y_i
 - In general, all the approaches have worse performance with smaller α

Ref: John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira, “*Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*”, ICML, 2001

Synthetic Data: CRF v.s. HMM



CRF - Summary

Problem 1:
Evaluation

$$F(x, y) = P(y|x) = \frac{\exp(w \cdot \phi(x, y))}{\sum_{y' \in \mathbb{Y}} \exp(w \cdot \phi(x, y'))}$$

Problem 2:
Inference

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(y|x) = \arg \max_{y \in \mathbb{Y}} w \cdot \phi(x, y)$$

Problem 3:
Training

$$w^* = \arg \max_w \prod_{n=1}^N P(\hat{y}^n | x^n)$$

$$w \rightarrow w + \eta \left(\phi(x^n, \hat{y}^n) - \sum_{y'} P(y'|x^n) \phi(x^n, y') \right)$$

Outline

Hidden Markov Model (HMM)



Conditional Random Field (CRF)



Structured Perceptron/SVM



Towards Deep Learning

Structured Perceptron

Problem 1:
Evaluation

$$F(x, y) = w \cdot \underline{\phi(x, y)}$$

The same as CRF

Problem 2:
Inference

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} w \cdot \phi(x, y)$$

Viterbi

Problem 3:
Training

$$\forall n, \forall y \in \mathbb{Y}, y \neq \hat{y}^n:$$

$$w \cdot \phi(x^n, \hat{y}^n) > w \cdot \phi(x^n, y)$$

$$\tilde{y}^n = \arg \max_y w \cdot \phi(x^n, y)$$

$$w \rightarrow w + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)$$

Structured Perceptron v.s. CRF

- Structured Perceptron

$$\tilde{y}^n = \arg \max_y w \cdot \phi(x^n, y)$$

$$w \rightarrow w + \underbrace{\phi(x^n, \hat{y}^n)}_{\text{Hard}} - \underbrace{\phi(x^n, \tilde{y}^n)}_{\text{Hard}}$$

Hard

- CRF

$$w \rightarrow w + \eta \left(\underbrace{\phi(x^n, \hat{y}^n)}_{\text{Soft}} - \underbrace{\sum_{y'} P(y'|x^n) \phi(x^n, y')}_{\text{Soft}} \right)$$

Soft

Structured SVM

Problem 1:
Evaluation

$$F(x, y) = w \cdot \underline{\phi(x, y)}$$

The same as CRF

Problem 2:
Inference

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} w \cdot \phi(x, y)$$

Viterbi

Problem 3:
Training

Consider margin and error:

Way 1. Gradient Descent

Way 2. Quadratic Programming
(Cutting Plane Algorithm)

Structured SVM – Error Function

- Error function: $\Delta(\hat{y}^n, y)$
 - $\Delta(\hat{y}^n, y)$: Difference between y and \hat{y}^n
 - Cost function of structured SVM is the upper bound of $\Delta(\hat{y}^n, y)$
 - Theoretically, $\Delta(y, \hat{y}^n)$ can be any function you like
 - However, you need to solve **Problem 2.1**
 - $\bar{y}^n = \underset{y}{\operatorname{argmax}} [\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)]$

Example

\hat{y} : A T T C G G G G A T

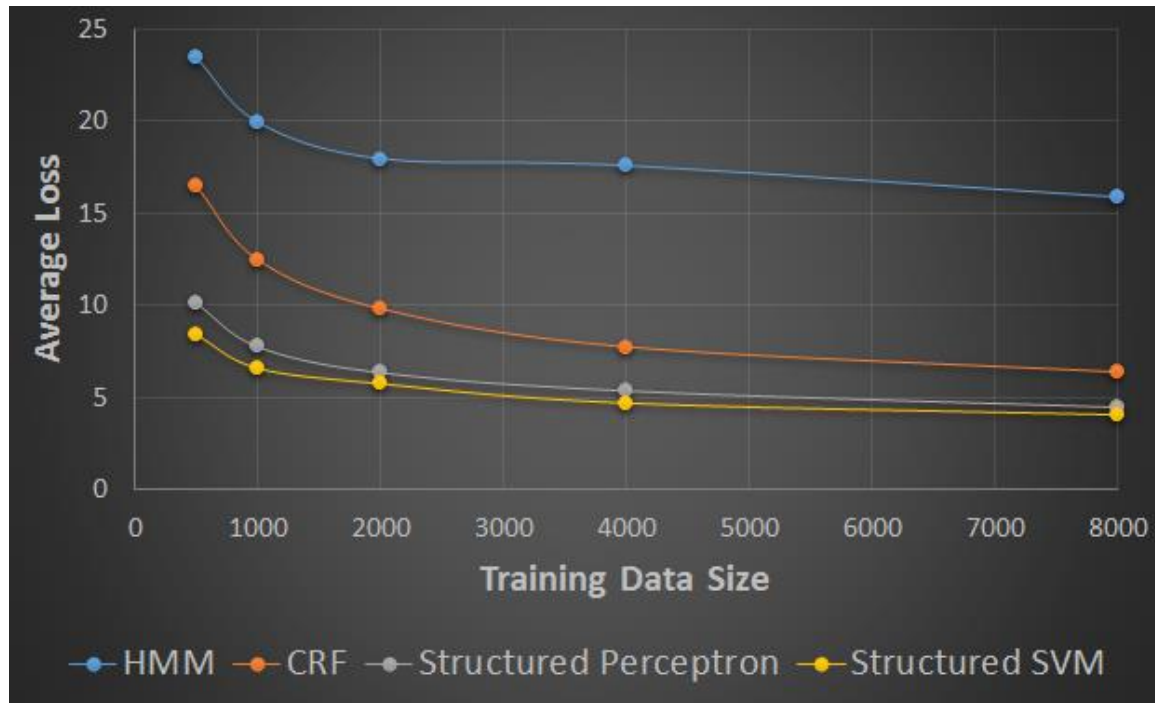
$\Delta(\hat{y}, y) = 3/10$ y : A T T A G G A G A A

In this case, problem 2.1 can be solved by Viterbi Algorithm

Performance of Different Approaches

POS Tagging

Ref: Nguyen, Nam, and Yunsong Guo.
"Comparisons of sequence labeling
algorithms and extensions." *ICML*, 2007.



Name Entity Recognition

Method	HMM	CRF	Perceptron	SVM
Error	9.36	5.17	5.94	5.08

Ref: Tsochantaridis, Ioannis, et al. "Large margin methods for structured and interdependent output variables." *Journal of Machine Learning Research*. 2005.

Outline

Hidden Markov Model (HMM)



Conditional Random Field (CRF)



Structured Perceptron/SVM



Towards Deep Learning

How about RNN?

- RNN, LSTM

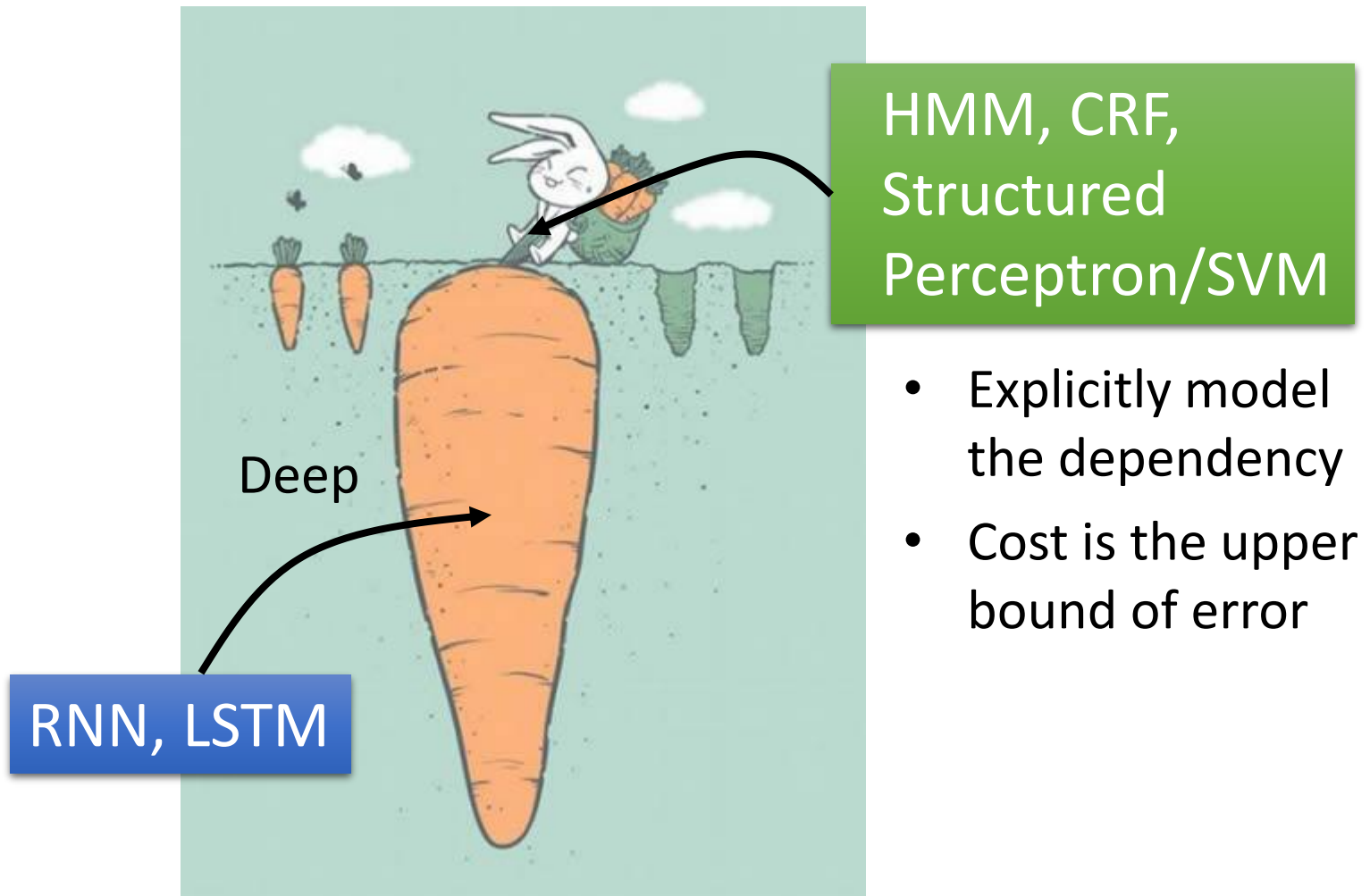
- Unidirectional RNN does not consider the whole sequence
- Cost and error not always related
- Deep 勝



- HMM, CRF, Structured Perceptron/SVM

- Using Viterbi, so consider the whole sequence 勝?
- How about Bidirectional RNN?
- Can explicitly consider the label dependency 勝
- Cost is the upper bound of error 勝

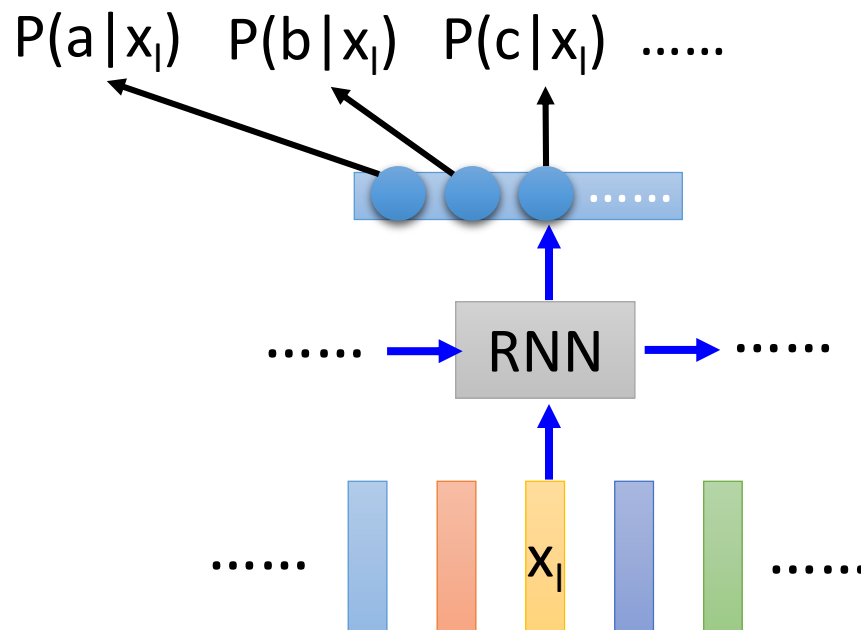
Integrated together



Integrated together

- Speech Recognition: CNN/RNN or LSTM/DNN + HMM

$$P(x, y) = P(y_1 | start) \prod_{l=1}^{L-1} P(y_{l+1} | y_l) P(end | y_L) \prod_{l=1}^L \underline{P(x_l | y_l)}$$

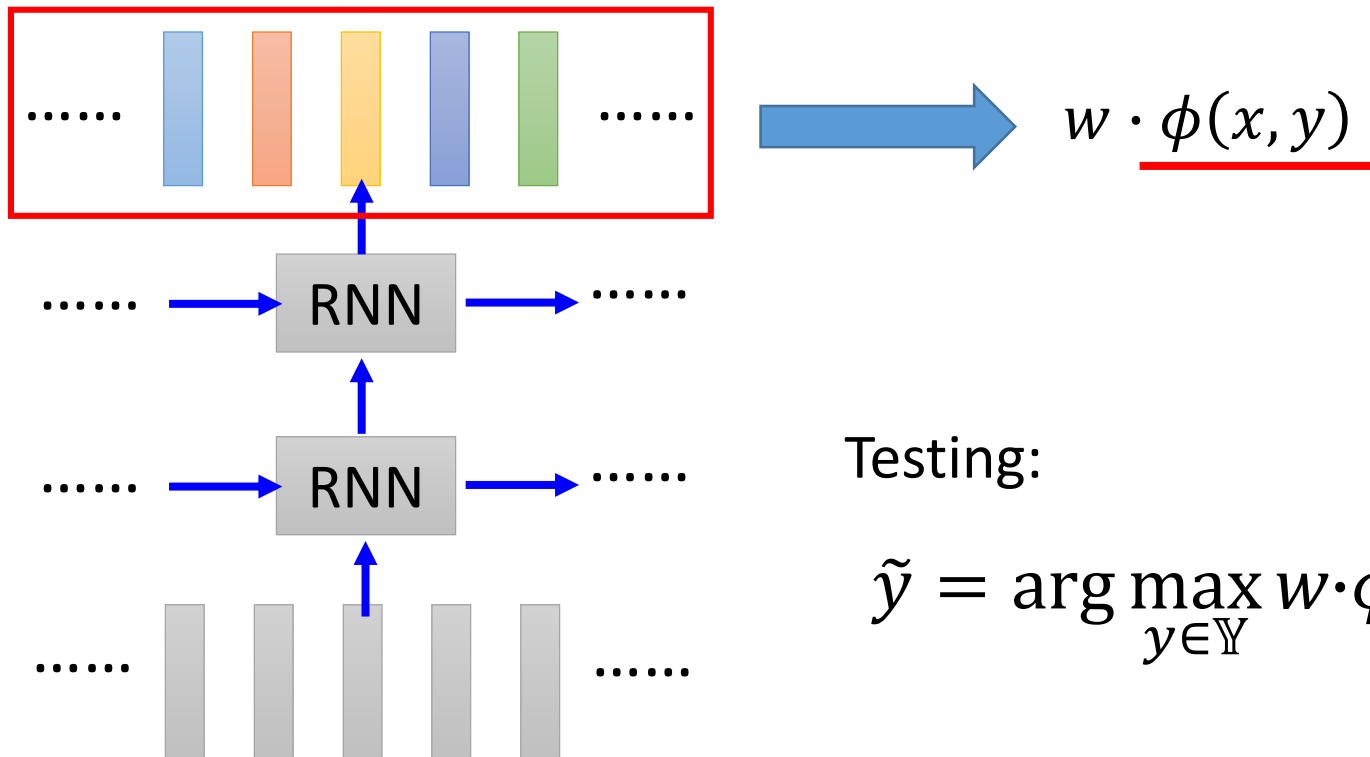


$$P(x_l | y_l) = \frac{P(x_l, y_l)}{P(y_l)}$$

$$= \frac{\text{RNN} \quad P(y_l | x_l) \cancel{P(x_l)}}{\text{Count} \quad P(y_l)}$$

Integrated together

- Semantic Tagging: Bi-directional RNN/LSTM + CRF/Structured SVM



Concluding Remarks

	Problem 1	Problem 2	Problem 3
HMM	$F(x, y) = P(x, y)$	Viterbi	Just count
CRF	$F(x, y) = P(y x)$	Viterbi	Maximize $P(\hat{y} x)$
Structured Perceptron	$F(x, y) = w \cdot \phi(x, y)$ (not a probability)	Viterbi	$F(x, \hat{y}) > F(x, y')$
Structured SVM	$F(x, y) = w \cdot \phi(x, y)$ (not a probability)	Viterbi	$F(x, \hat{y}) > F(x, y')$ with margins

The above approaches can combine with deep learning to have better performance.

Acknowledgement

- 感謝 曹燁文 同學於上課時發現投影片上的錯誤
- 感謝 **Ryan Sun** 來信指出投影片上的錯誤

Appendix

CRF - Training

$$O^n(w) = \log \frac{\exp(w \cdot \phi(x^n, \hat{y}^n))}{Z(x^n)} \quad Z(x^n) = \sum_{y'} \exp(w \cdot \phi(x^n, y'))$$
$$= \underline{w \cdot \phi(x^n, \hat{y}^n)} - \log Z(x^n)$$

$$\frac{\partial O^n(w)}{\partial w_{s,t}} = \underline{N_{s,t}(x^n, \hat{y}^n)}$$



The number of word t labeled as s in (x^n, \hat{y}^n)

The value of the dimension in $\phi(x^n, \hat{y}^n)$ corresponding to $w_{s,t}$.

$$w \cdot \phi(x^n, \hat{y}^n) = \sum_{s,t} w_{s,t} \cdot N_{s,t}(x^n, \hat{y}^n) + \sum_{s,s'} w_{s,s'} \cdot N_{s,s'}(x^n, \hat{y}^n)$$

CRF - Training

$$O^n(w) = \log \frac{\exp(w \cdot \phi(x^n, \hat{y}^n))}{Z(x^n)} \quad Z(x^n) = \sum_{y'} \exp(w \cdot \phi(x^n, y'))$$

$$= \underline{w \cdot \phi(x^n, \hat{y}^n)} - \underline{\log Z(x^n)}$$

$$\frac{\partial O^n(w)}{\partial w_{s,t}} = \underline{N_{s,t}(x^n, \hat{y}^n)} - \frac{1}{\underline{Z(x^n)}} \underline{\frac{\partial Z(x^n)}{\partial w_{s,t}}}$$

$$= \sum_{y'} \left[\frac{\exp(w \cdot \phi(x^n, y'))}{Z(x^n)} \right] N_{s,t}(x^n, y') = \sum_{y'} \left[P(y'|x^n) \right] N_{s,t}(x^n, y')$$

$$\underline{\frac{\partial Z(x^n)}{\partial w_{s,t}}} = \sum_{y'} \exp(w \cdot \phi(x^n, y')) N_{s,t}(x^n, y')$$

CRF v.s. HMM

- Define $\phi(x, y)$ you like
 - For example, besides the features just described, there are some useful extra features in POS tagging.
 - Number of times a capitalized word is labeled as Noun
 - Number of times a word end with **ing** is labeled as Noun
- Can you consider this kind of features by HMM? Too sparse...

$$P(x_i = A, x_i \text{ is capitalized}, x_i \text{ end with } \mathbf{ing}, \dots | y_i = N)$$

Method 1:

$$P(x_i = A | y_i = N) P(x_i \text{ is capitalized} | y_i = N) \dots$$

Inaccurate assumption

Method 2. Give the distribution some assumptions?