Sequence Labeling Problem

Sequence Labeling

$$f : X \rightarrow Y$$

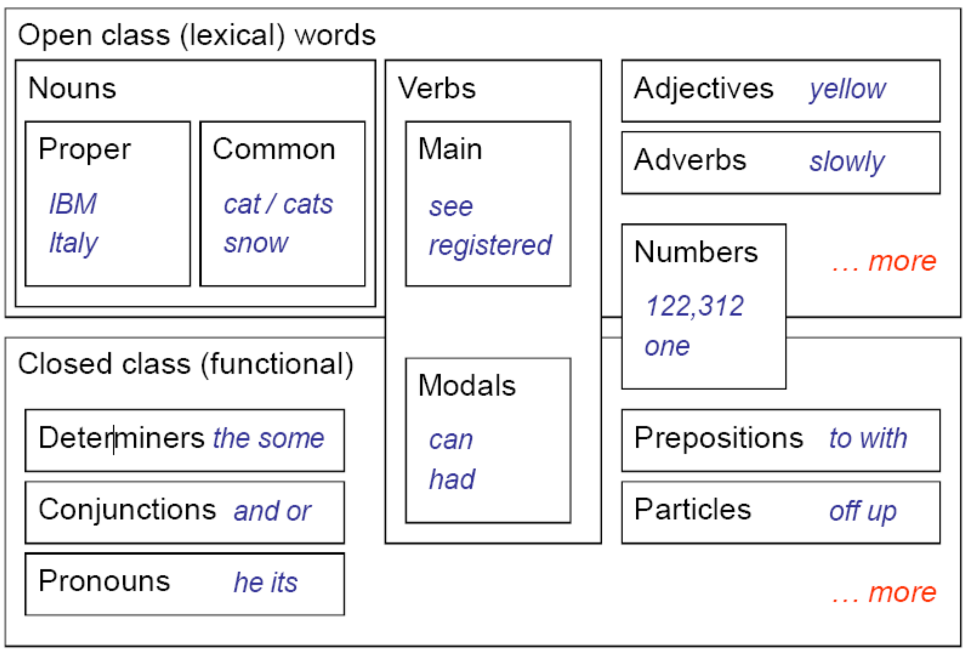
Sequence Sequence

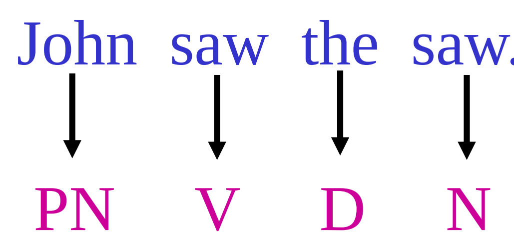
$$\begin{array}{ccccccc}
x: & x_1 & x_2 & x_3 & \cdots\cdots & x_L & \text{(sequence)} \\
\downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \\
y: & y_1 & y_2 & y_3 & \cdots\cdots & y_L & \text{(sequence)}
\end{array}$$

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.





  - 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?

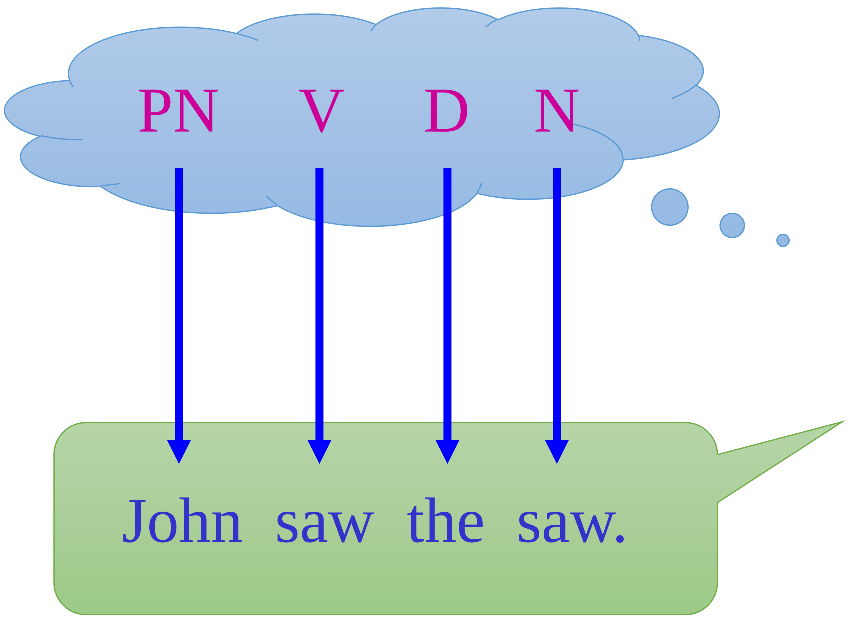
Just the assumption of HMM

Step 1

- Generate a POS sequence
- Based on the grammar

Step 2

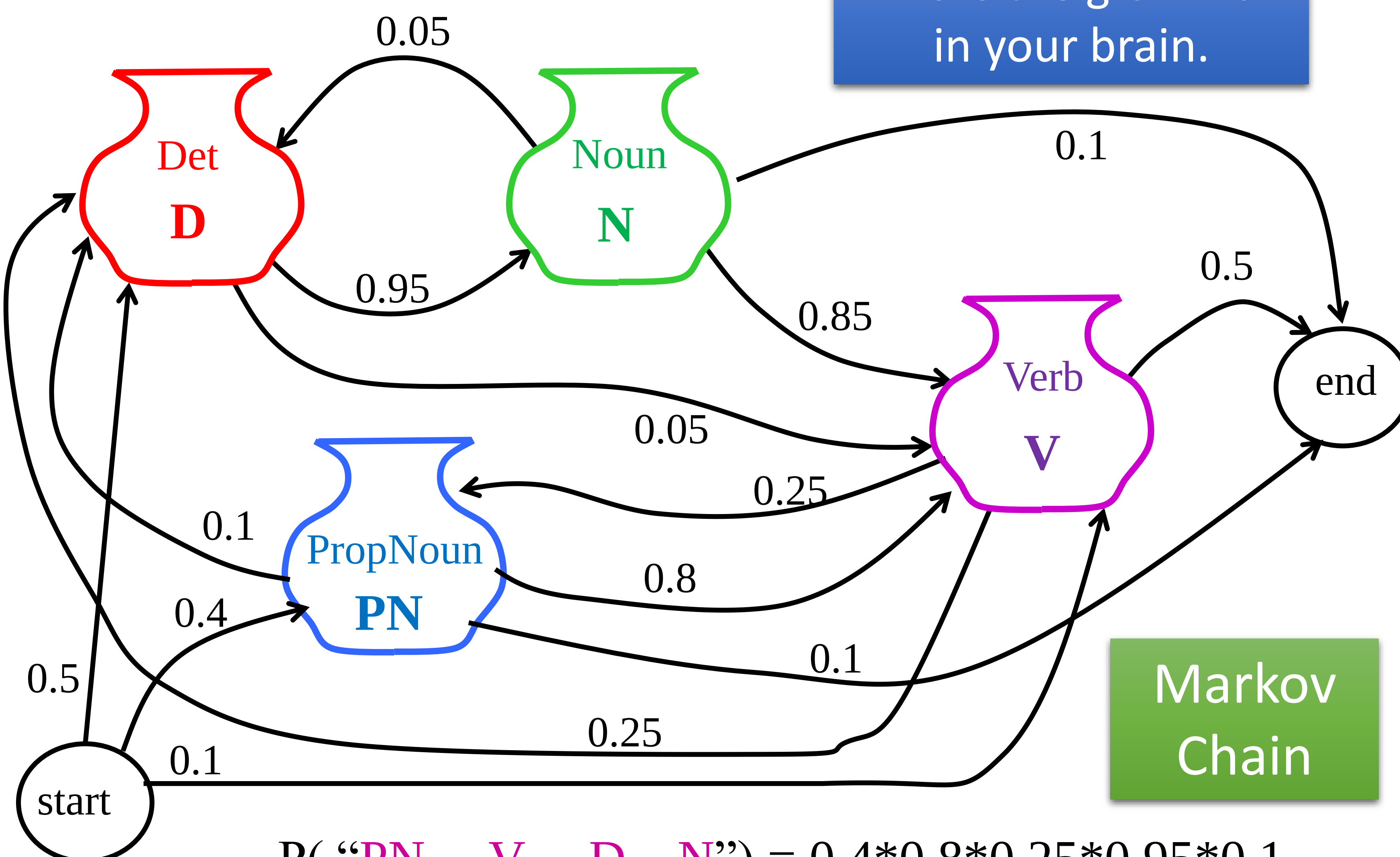
- Generate a sentence based on the POS sequence
- Based on a dictionary



[Slide credit: Raymond J. Mooney]

HMM – Step 1

This is the grammar in your brain.

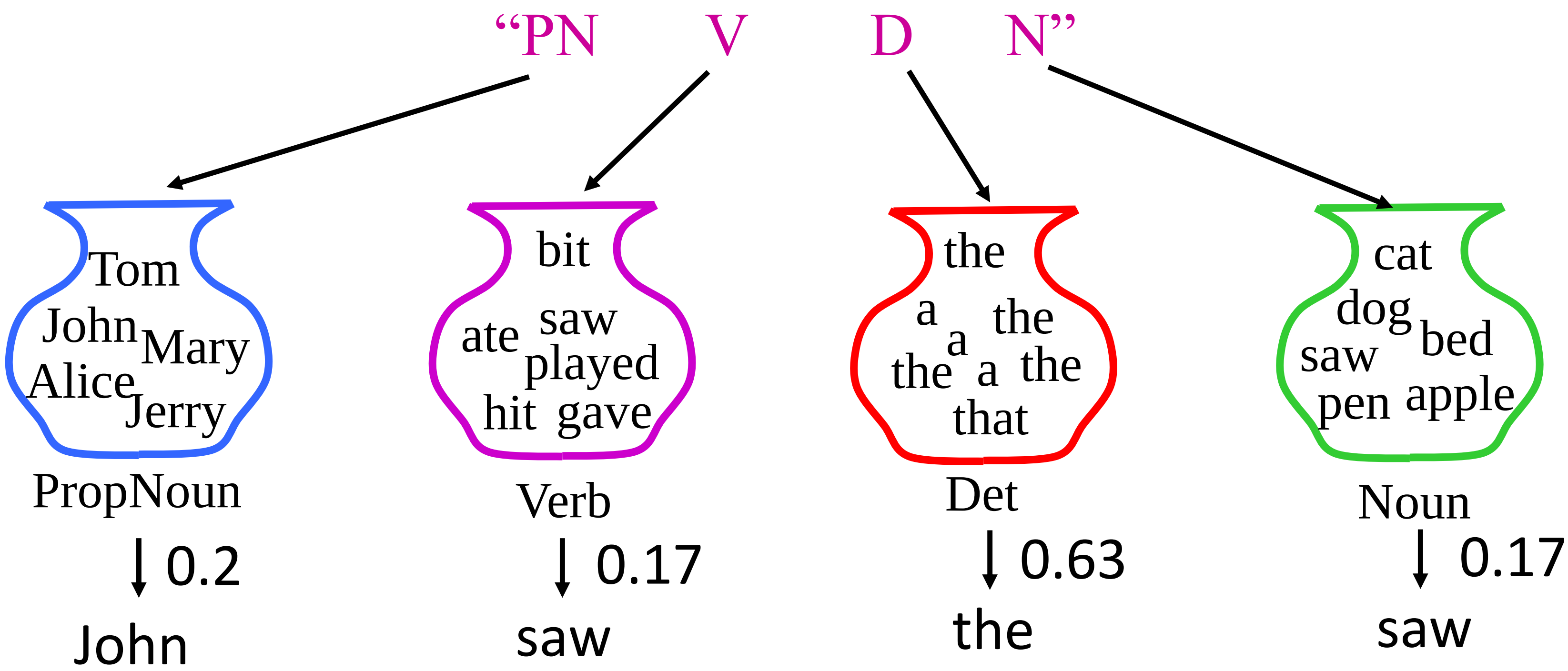


Markov Chain

P(“PN V D N”) = 0.4*0.8*0.25*0.95*0.1

[Slide credit: Raymond J. Mooney]

HMM – Step 2



P("John saw the saw" | "PN V D N")
= 0.2*0.17*0.63*0.17

HMM

x: John saw the saw.

y: ***start*** $\rightarrow$ PN $\rightarrow$ V $\rightarrow$ D $\rightarrow$ N $\rightarrow$ ***end***

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

$$P(y) = P(PN|start) \times P(V|PN) \times P(D|V) \times P(N|D)$$

$$P(x|y) = P(John|PN) \times P(saw|V) \times P(the|D) \times P(saw|N)$$

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(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

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

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

(s and s' are tags)

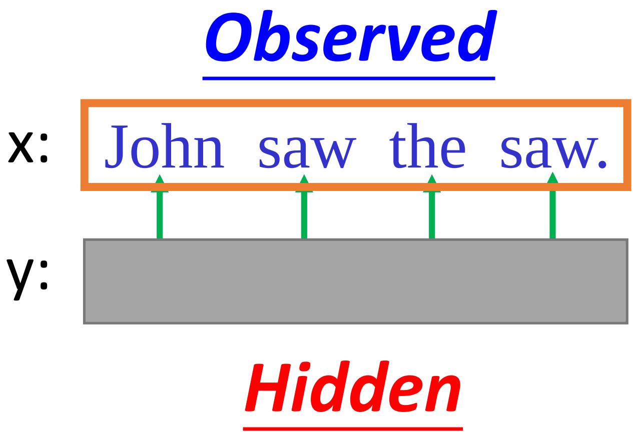
$$\underline{P(x_l = t|y_l = s)} = \frac{count(s \to t)}{count(s)}$$

(s is tag, and t is word)

So simple ☺

HMM – How to do POS Tagging?

- We can compute P(x,y)



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

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

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

Task: given x, find 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

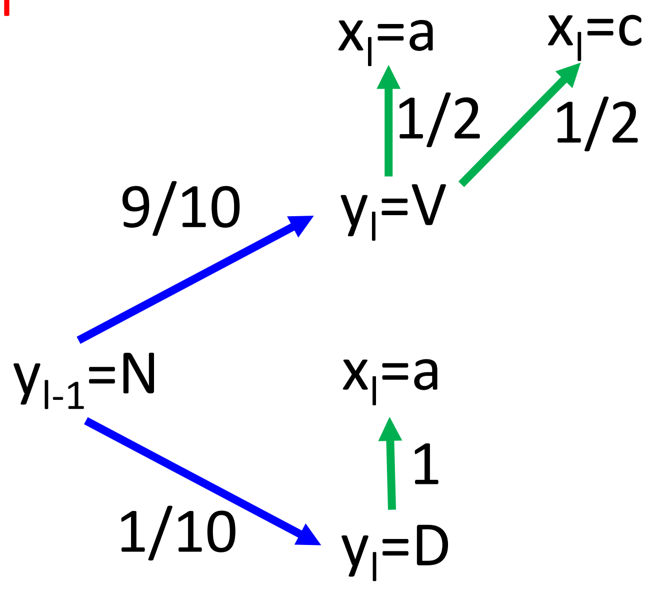
- Inference:

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

- 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 P(D|N)=1/10 ……

Emission probability:

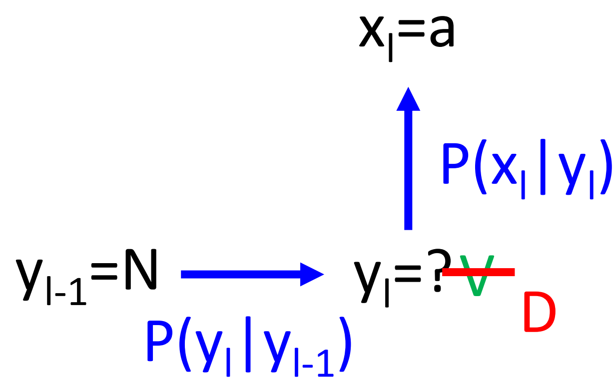
P(a|V)=1/2 P(a|D)=1 ……

$x_l=a$

$\uparrow P(x_l|y_l)$

$y_{l-1}=N \xrightarrow{P(y_l|y_{l-1})} y_l=?$ V

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?

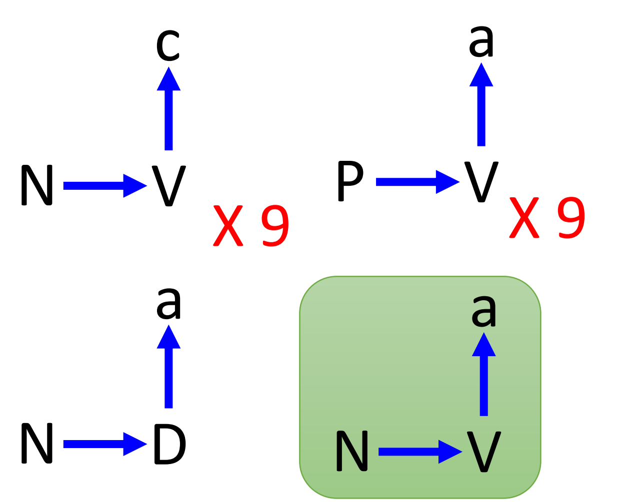
not necessarily small

Transition probability:

P(V|N)=9/10 P(D|N)=1/10 ……

Emission probability:

P(a|V)=1/2 P(a|D)=1 ……



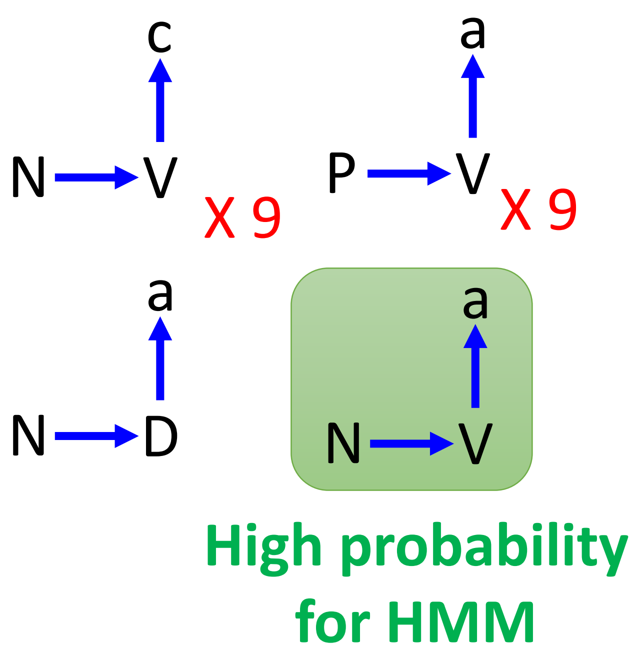
High probability for HMM

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

$$\mathrm{P}(x,y) \propto exp\left(w \cdot \phi(x,y)\right)$$

- $\phi(x,y)$ is a feature vector. What does it look like?
- w is a weight vector to be learned from training data
- $exp\left(w \cdot \phi(x,y)\right)$ is always positive, can be larger than 1

$$P(y|x) = \frac{P(x,y)}{\sum_{y'} P(x,y')} \qquad \mathrm{P}(x,y) = \frac{exp\left(w \cdot \phi(x,y)\right)}{R}$$

$$= \frac{exp\left(w \cdot \phi(x,y)\right)}{\sum_{y' \in \mathbb{Y}} exp\left(w \cdot \phi(x,y')\right)} = \frac{exp\left(w \cdot \phi(x,y)\right)}{Z(x)}$$

P(x,y) for CRF

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

very different from HMM?

In HMM:

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

$$logP(x,y)$$

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

$$+ \sum_{l=1}^{L} logP(x_l|y_l)$$

P(x,y) for CRF

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

$$\boxed{+ \sum_{l=1}^{L} logP(x_l|y_l)}$$

Log probability of word t given tag s

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

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

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)

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

$$= \cancel{logP(the|D)} + logP(dog|N) + logP(ate|V)$$
$$+ \cancel{logP(the|D)} + logP(homework|N)$$

$$= \cancel{logP(the|D) \times 2} + logP(dog|N) \times 1 + logP(ate|V) \times 1$$
$$+ logP(homework|N) \times 1$$

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

P(x,y) for CRF

$$logP(x,y)$$

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

$$+ \sum_{l=1}^{L} logP(x_l|y_l)$$

$$logP(y_1|start) = \sum_{s} logP(s|start) \times N_{start,s}(x,y)$$

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

$$logP(end|y_L) = \sum_{s} logP(end|s) \times N_{s,end}(x,y)$$

P(x,y) for CRF

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

$$
= \begin{bmatrix} \vdots \\ \log P(t|s) \\ \vdots \\ \vdots \\ \log P(s|start) \\ \vdots \\ \vdots \\ \log P(s'|s) \\ \vdots \\ \vdots \\ \log P(end|s) \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ N_{s,t}(x,y) \\ \vdots \\ \vdots \\ N_{start,s}(x,y) \\ \vdots \\ \vdots \\ N_{s,s'}(x,y) \\ \vdots \\ \vdots \\ N_{s,end}(x,y) \\ \vdots \end{bmatrix}
$$

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

$$\mathrm{P}(x,y) = exp\big(w \cdot \phi(x,y)\big)$$

P(x,y) for CRF

$$\mathrm{P}(x,y) \cancel{=} \propto exp\left(w \cdot \phi(x,y)\right)$$

However, we do not give w any constraints during training

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

$w_{s,t} \rightarrow logP(x_i = t|y_i = s)$

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

$w_{start,s} \rightarrow logP(s|start)$

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

$w_{s,s'} \rightarrow logP(y_i = s'|y_{i-1} = s)$

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

$w_{s,end} \rightarrow logP(end|s)$

…… ……

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| X |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)

	Part 2	Value
$N_{D,D}(x,y) \rightarrow$	D, D	0
$N_{D,N}(x,y) \rightarrow$	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
  - $\boxed{\text{Part 2: relations between tags}}$

If there are |S| possible tags,
|S| X |S| + 2 |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

CRF – Training Criterion

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

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

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

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

Maximize what we observe

Minimize what we don’t observe

CRF – Gradient Ascent

<u>Gradient descent</u>

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

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

Opposite direction of the gradient

<u>Gradient Ascent</u>

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

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

The same direction of the gradient

CRF - Training

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

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

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

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

After some math ……

Can be computed by Viterbi algorithm as well

$$\frac{\partial O^n(w)}{\partial w_{s,t}} = \underline{N_{s,t}(x^n, \hat{y}^n)} - \underline{\sum_{y'} P(y'|x^n)\, N_{s,t}\,(x^n, y')}$$

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 $w_{s,t}$

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

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 $(x^n, \hat{y}^n)$

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

CRF – Inference

- Inference

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

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

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

CRF v.s. HMM

- CRF: increase $P(x, \hat{y})$, decrease $P(x, y')$
  - To obtain correct results …

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

HMM does not do that

CRF more likely to achieve that than HMM

$N \rightarrow V \xrightarrow{\uparrow} c$ X 9

$P \rightarrow V \xrightarrow{\uparrow} a$ X 9

$N \rightarrow D \xrightarrow{\uparrow} a$

HMM:

P(V|N)=9/10

P(D|N)=1/10

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

P(a|D)=1

CRF:

$x_i = a$

$P(x_i|y_i)$

$y_{i-1}=N \xrightarrow{P(y_i|y_{i-1})} y_i=?$

HMM:V

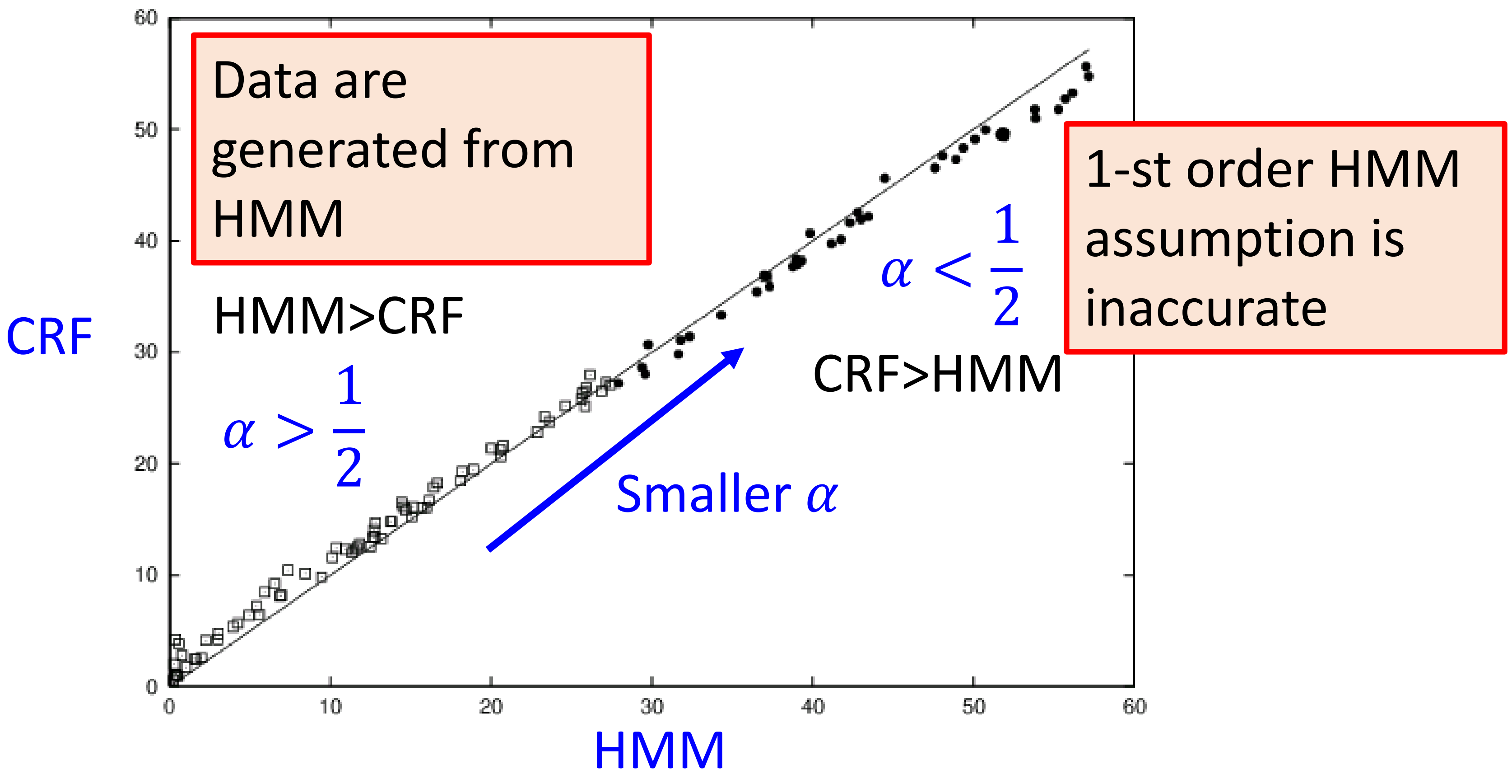
CRF:D

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

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

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

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

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

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***

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

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

Hard

- ***CRF***

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

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***
    - $\bar{y}^n = arg\max_{y}[\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)]$

Example

$\Delta(\hat{y}, y) = 3/10$

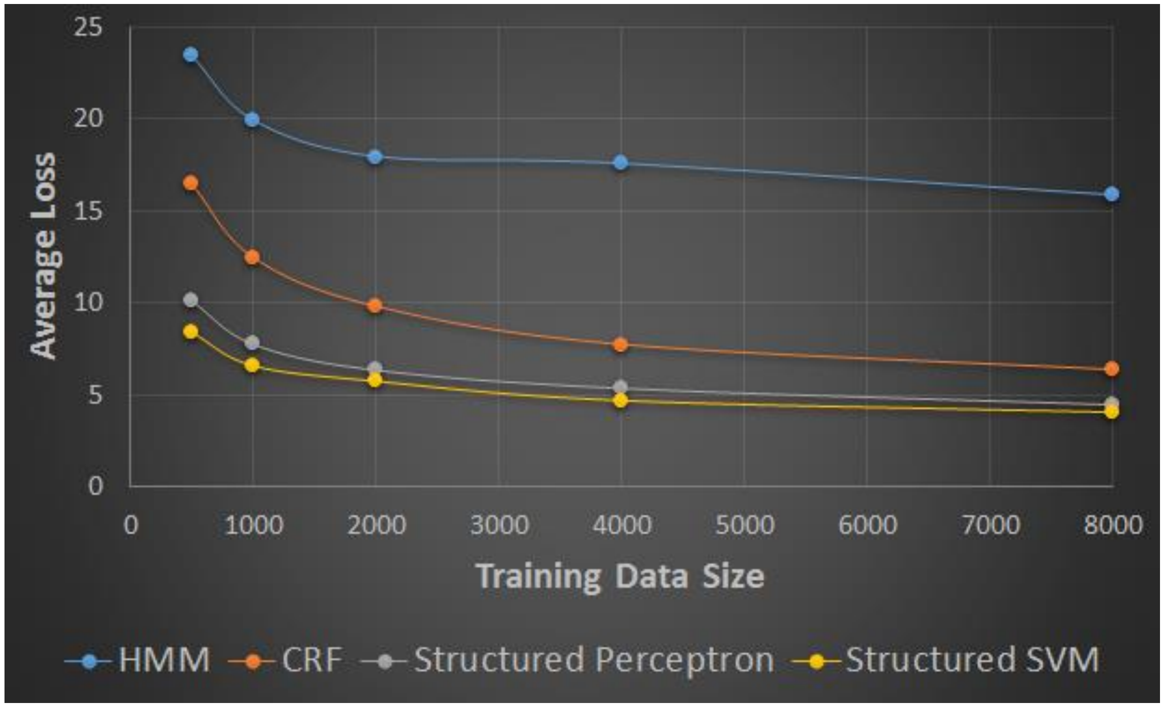
$\hat{y}$:	A	T	T	C	G	G	G	G	A	T
y:	A	T	T	A	G	G	A	G	A	A

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?

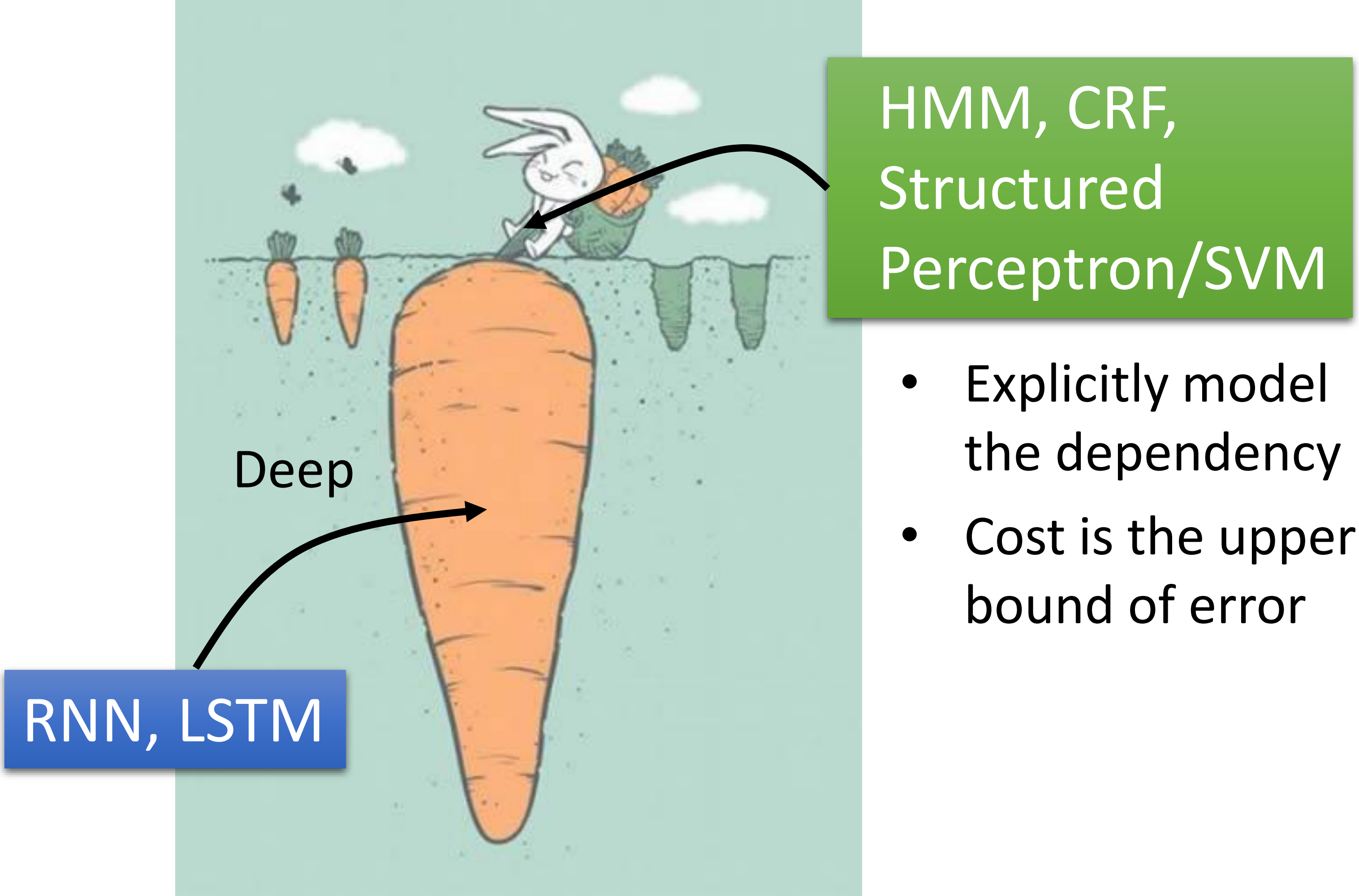
- RNN, LSTM
  - Unidirectional RNN does not consider the whole sequence
  - Cost and error not always related
  - Deep 

- HMM, CRF, Structured Perceptron/SVM
  - Using Viterbi, so consider the whole sequence 

?
    - How about Bidirectional RNN?
  - Can explicitly consider the label dependency 

  - Cost is the upper bound of error 

Integrated together



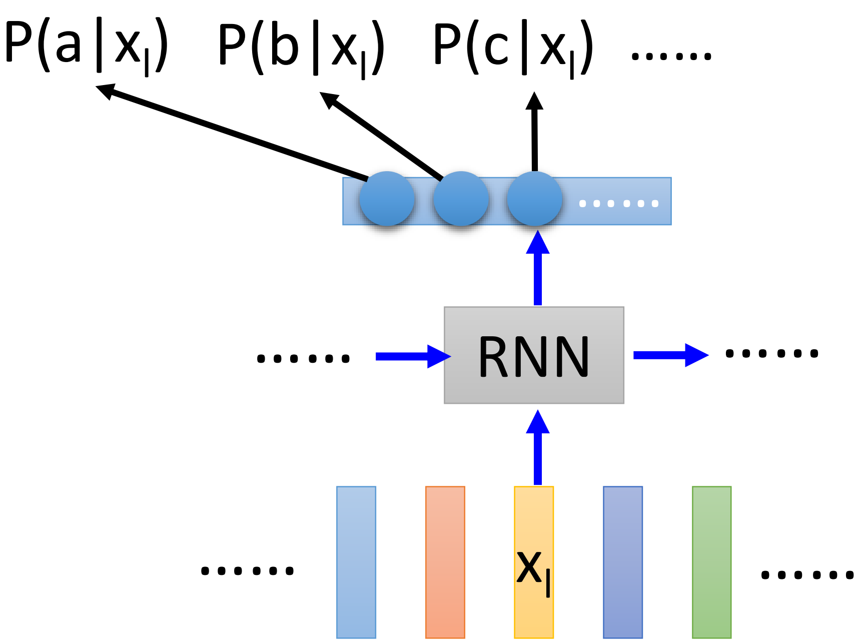
- Explicitly model the dependency
- Cost is the upper bound of error

http://photo30.bababian.com/upload1/20100415/42E9331A6580A46A5F89E98638B8FD76.jpg

Integrated together

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

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



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

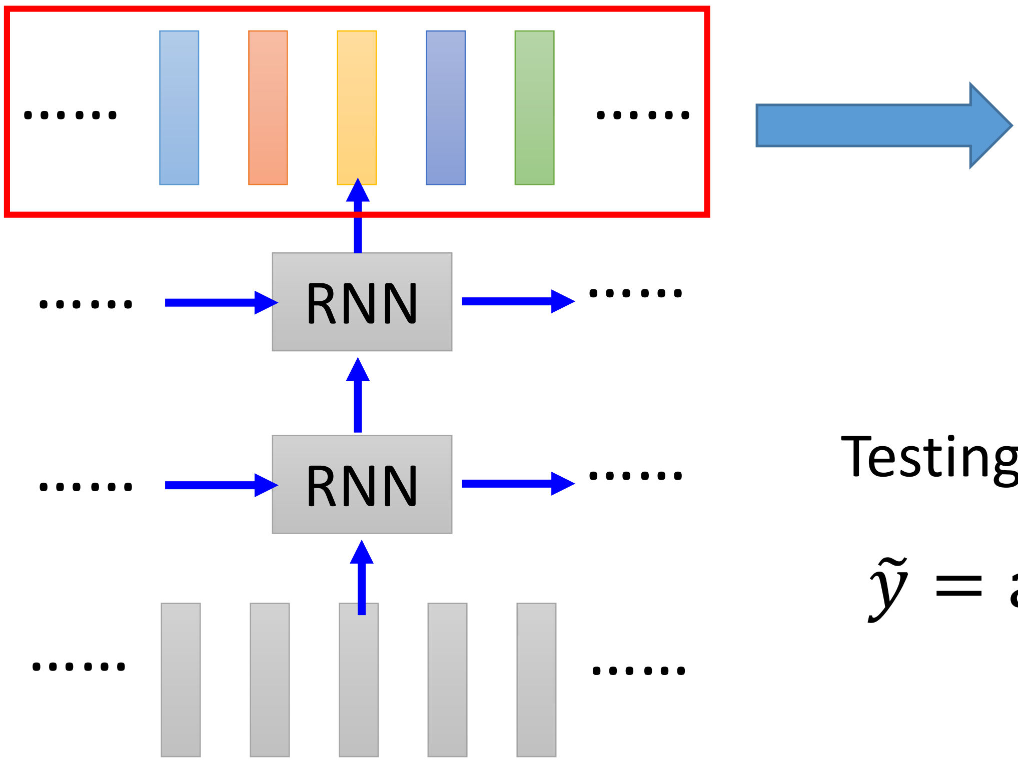
RNN

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

Count

Integrated together

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



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

Testing:

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

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 曹爗文 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

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

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

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

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

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

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

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

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

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

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…

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

Method 1:

$$P(x_i = A | y_i = N) P(x_i \ is \ capitalized | y_i = N) \ldots\ldots$$

Inaccurate assumption

Method 2. Give the distribution some assumptions?