PoS tagging with Maximum entropy models

Miguel A. Alonso

Departamento de Computación, Facultad de Informática, Universidade da Coruña

Outline

Maximum entropy models: MaxEnt

- It is a probabilistic machine learning framework
- Based on **multinomial logistic regression**
- When used to classify data sequences, it usually takes the form of a maximum entropy Markov model or MEMM
- In the case of PoS tagging:
  - the sequence to classify are the words of a text
  - the goal is to assign a PoS tag to each word

Maximum entropy models: MaxEnt

- MaxEnt belongs to the family of **exponential** or **log-linear** classifiers
  - extracts a set of (textbf features) from the input
  - combines them **linear**ly
  - uses the result as an **exponent**
- Given an entry x with features f_i weighted by w_i, the probability of assigning x the class c is

$$P(c \mid x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i\right)$$

  where Z is a normalization fact in order to sum up 1
- When the result belongs to a discrete set, we talk about **classification**, when it is a real set, we talk about **regression**

Outline

Example of lineal regression

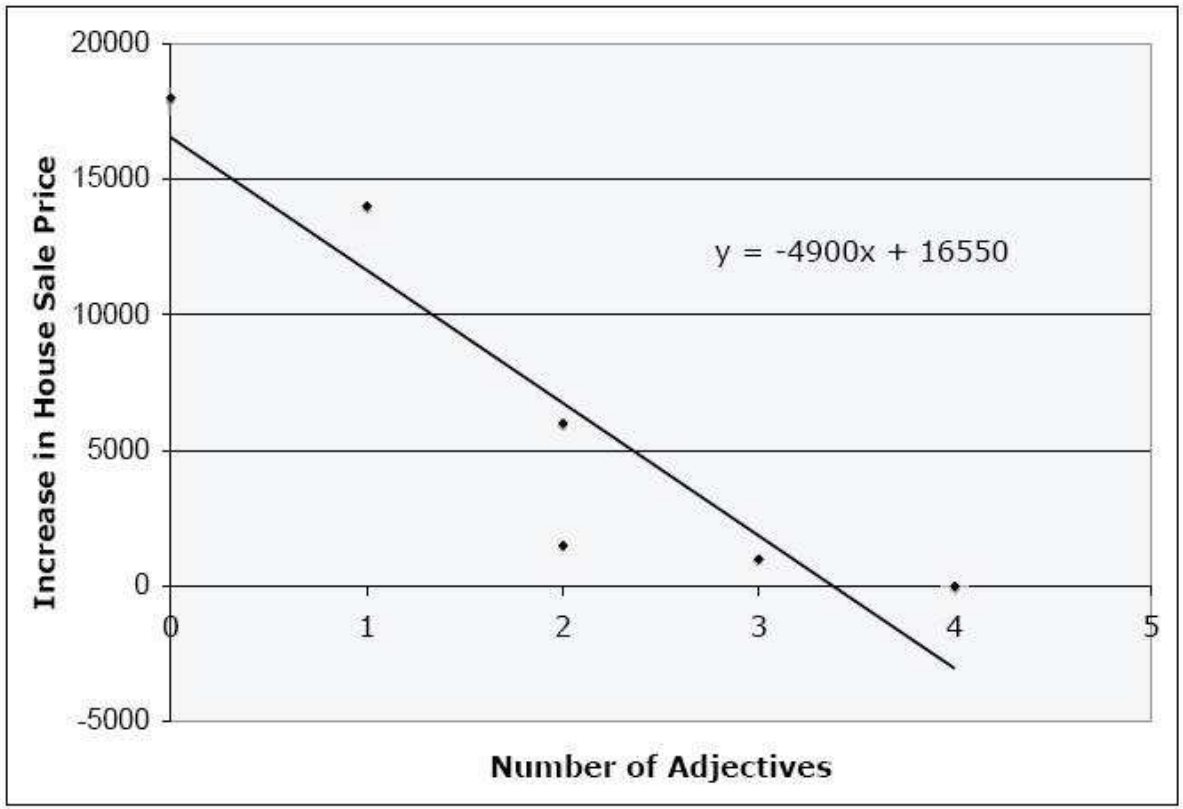
In the book *Freakonomics* it is suggested that the words of home-for-sale advertisements can help predict if they will sell for more or less of the advertised price. The hypothesis is that words of vague meaning (beautiful, cozy, bright, ...) are used to mask the absence of concrete positive properties.

Let us suppose the following data:

# of Vague Adjectives	Amount House Sold Over Asking Price
4	0
3	$1000
2	$1500
2	$6000
1	$14000
0	$18000

Example of lineal regression

from which we can get the following **regression line**:



$price = w_0 + w_1 * f_1$ where $w_0 = 16.550$ and $w_1 = -4.900$ and $f_1 = \#\text{adjective}$

Multiple linear regression

- When there is more than one feature, we talk about **multiple linear regression**

$$y = \sum_{i=0}^{N} w_i \ f_i$$

where in general $w_0 = 1$, which can be written as the scalar product of vectors $\vec{w}$ and $\vec{f}$

- The learning of weights $\vec{w}$ is usually done by minimizing the mean squared error between the predicted and observed values

Logistic regression

- We are not interested in y but in $P(y)$, the probability of y, where y is a set of discrete values (probabilistic classification)
- Consequently, we would like to calculate $P(y = \text{true} \mid x) = \sum_{i=0}^{N} w_i f_i$ ... but it turns out that $\sum_{i=0}^{N} w_i f_i$ yields values between $-\infty$ and ∞
- Solution: make $\sum_{i=0}^{N} w_i f_i$ a ratio between two probabilities (and not a probability)

Regresión logística

- The ratio we use is the odds of a probability:

$$\frac{P(y = \text{true} \mid x)}{1 - P(y = \text{true} \mid x)} = \sum_{i=0}^{N} w_i f_i$$

- The left part can vary between 0 and ∞. In order to make it vary between $-\infty$ and ∞ we apply logarithms:

$$\ln\left(\frac{P(y = \text{true} \mid x)}{1 - P(y = \text{true} \mid x)}\right) = \sum_{i=0}^{N} w_i f_i$$

- **Logistic regression is a regression model that uses a linear function to estimate the logarithm of the odds of a probability**

How to get $P(y = \text{true} \mid x)$

$$\ln\left(\frac{P(y = \text{true} \mid x)}{1 - P(y = \text{true} \mid x)}\right) = \sum_{i=0}^{N} w_i f_i$$

$$\frac{P(y = \text{true} \mid x)}{1 - P(y = \text{true} \mid x)} = \exp\left(\sum_{i=0}^{N} w_i f_i\right)$$

$$P(y = \text{true} \mid x) = (1 - P(y = \text{true} \mid x)) \exp\left(\sum_{i=0}^{N} w_i f_i\right)$$

$$P(y = \text{true} \mid x) = \exp\left(\sum_{i=0}^{N} w_i f_i\right) - P(y = \text{true} \mid x) \exp\left(\sum_{i=0}^{N} w_i f_i\right)$$

$$P(y = \text{true} \mid x) + P(y = \text{true} \mid x) \exp\left(\sum_{i=0}^{N} w_i f_i\right) = \exp\left(\sum_{i=0}^{N} w_i f_i\right)$$

How to get $P(y = \text{true} \mid x)$

$$P(y = \text{true} \mid x)(1 + \exp\left(\sum_{i=0}^{N} w_i f_i\right) = \exp\left(\sum_{i=0}^{N} w_i f_i\right)$$

$$P(y = \text{true} \mid x) = \frac{\exp\left(\sum_{i=0}^{N} w_i f_i\right)}{1 + \exp\left(\sum_{i=0}^{N} w_i f_i\right)}$$

Therefore

$$P(y = \text{false} \mid x) = \frac{1}{1 + \exp\left(\sum_{i=0}^{N} w_i f_i\right)}$$

so that $P(y = \text{true} \mid x) + P(y = \text{false} \mid x) = 1$

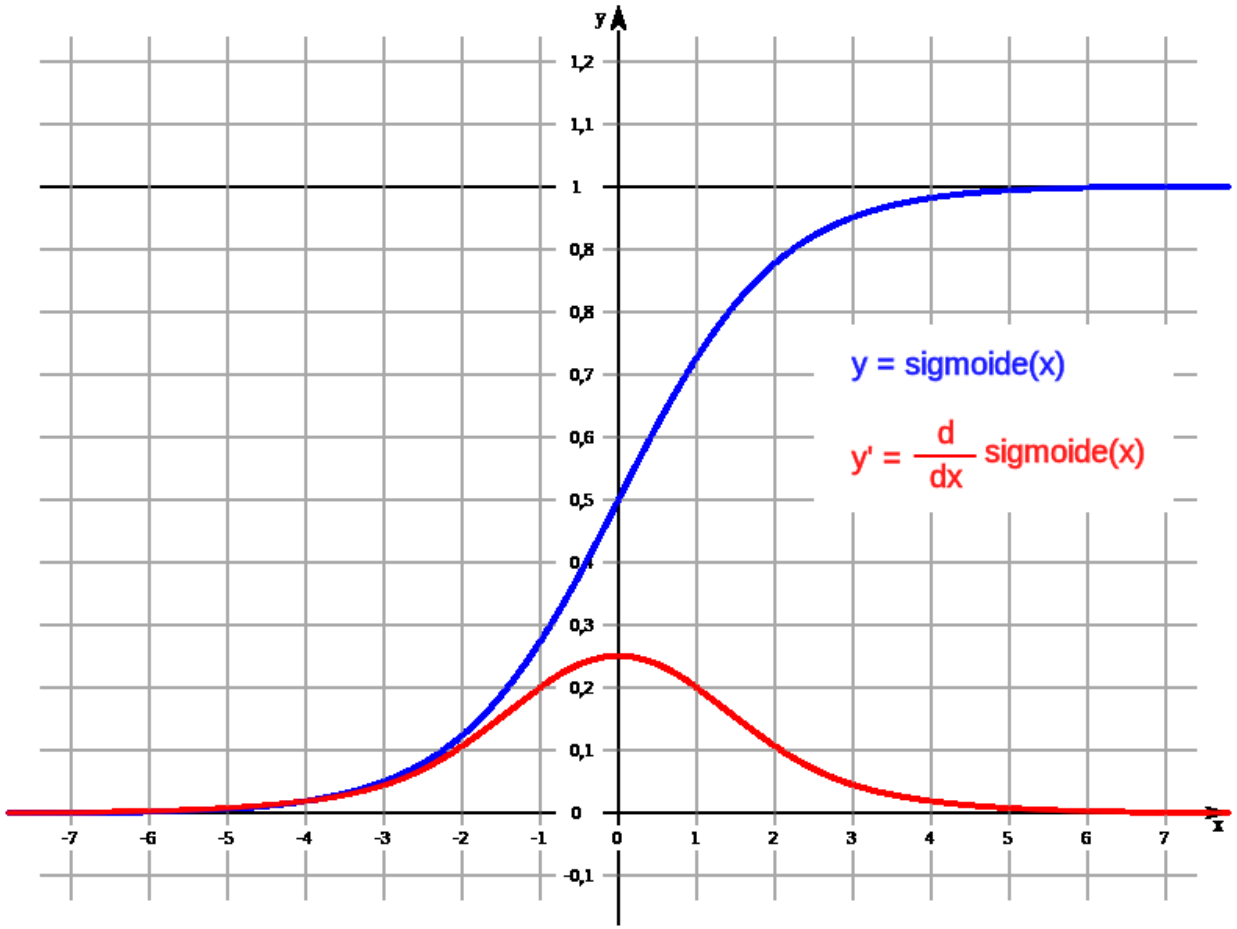
How to get $P(y = \text{true} \mid x)$

Let us take the expression $P(y = \text{true} \mid x) = \frac{\exp\left(\sum_{i=0}^{N} w_i f_i\right)}{1+\exp\left(\sum_{i=0}^{N} w_i f_i\right)}$, dividing numerator and denominator by $\exp\left(-\sum_{i=0}^{N} w_i f_i\right)$ we get

$$P(y = \text{true} \mid x) = \frac{1}{1 + \exp\left(-\sum_{i=0}^{N} w_i f_i\right)}$$

How to get $P(y = \text{true} \mid x)$

that is now in the form of a **logistic function** $\frac{1}{1+e^{-x}}$



How to get $P(y = \text{true} \mid x)$

and therefore:

$$P(y = \text{false} \mid x) = \frac{\exp\left(-\sum_{i=0}^{N} w_i f_i\right)}{1 + \exp\left(-\sum_{i=0}^{N} w_i f_i\right)}$$

Learning in logistic regression

- It is solved through complex mathematical techniques (non-linear programming) called textbf convex optimization
- We often use algorithms such as L-BFGS, ascending gradient algorithms, conjugate gradient algorithms, iterative scaling algorithms, ...
- The w_i weights are usually smoothed by **regularization** in order to penalize large weights

Outline

Classification with logistic regression

- The result will be the class *true* if $P(y = \text{true} \mid x) > P(y = \text{false} \mid x)$
- equivalently, if $\frac{P(y=\text{true}|x)}{P(y=\text{false}|x)} > 1$
- As $\frac{P(y=\text{true}|x)}{1-P(y=\text{true}|x)} = \exp(\sum_{i=0}^{N} w_i f_i)$
- it is then equivalent to see if $\exp(\sum_{i=0}^{N} w_i f_i) > 1$
- which implies to see if it is satisfied

$$\sum_{i=0}^{N} w_i f_i > 0$$

Multinomial logistic regression

Applied when we want to classify an observation into **more than two classes**

$$P(c \mid x) = \frac{1}{Z} \exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)$$

where we make the weights of f_i depend on the class $c \in C$ and

$$Z = \sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)$$

resulting

$$P(c \mid x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)}$$

Multinomial logistic regression for PoS tagging

- In PoS tagging the features f_i are not real but discrete
- More specifically, they are Boolean, indicating whether a property is present or not
- We denote by $f_i(c, x)$ the **indicator function** that tells us if a feature i is present for the class c in x.

$$P(c \mid x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i(c, x)\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i(c, x)\right)}$$

- The advantage with respect to other models is that the indicator functions can refer to "practically anything"

Example of non-sequential classification

Secretariat/NNP is/BEZ expected/VBN to/TO race/?? tomorrow/RB

$$f_1(c,x) = \left\{ \begin{array}{l} 1 \text{ si } w_i = \text{"race"} \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{array} \right\}$$

$$f_2(c,x) = \left\{ \begin{array}{l} 1 \text{ si } t_{i-1} = \text{TO} \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{array} \right\}$$

$$f_3(c,x) = \left\{ \begin{array}{l} 1 \text{ si suffix}(w_i) = \text{"ing"} \ \& \ c = \text{VBG} \\ 0 \text{ otherwise} \end{array} \right\}$$

$$f_4(c,x) = \left\{ \begin{array}{l} 1 \text{ si is\_lower\_case}(w_i) \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{array} \right\}$$

$$f_5(c,x) = \left\{ \begin{array}{l} 1 \text{ si } w_i = \text{"race"} \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{array} \right\}$$

$$f_6(c,x) = \left\{ \begin{array}{l} 1 \text{ si } t_{i-1} = \text{TO} \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{array} \right\}$$

Example of non-sequential classification

Given the current input x ="race" and supposing the following weights:

		f1	f2	f3	f4	f5	f6
VB	**f**	0	1	0	1	1	0
VB	**w**		.8		.01	.1	
NN	**f**	1	0	0	0	0	1
NN	**w**	.8					-1.3

$$P(NN \mid x) = \frac{e^{0.8}e^{-1.3}}{e^{0.8}e^{-1.3} + e^{0.8}e^{0.01}e^{0.1}} = 0.20$$

$$P(VB \mid x) = \frac{e^{0.8}e^{0.01}e^{0.1}}{e^{0.8}e^{-1.3} + e^{0.8}e^{0.01}e^{0.1}} = 0.80$$

When using MaxEnt for classification we obtain a probability distribution on the classes

Outline

Sequential classification: HMM vs. MEMM

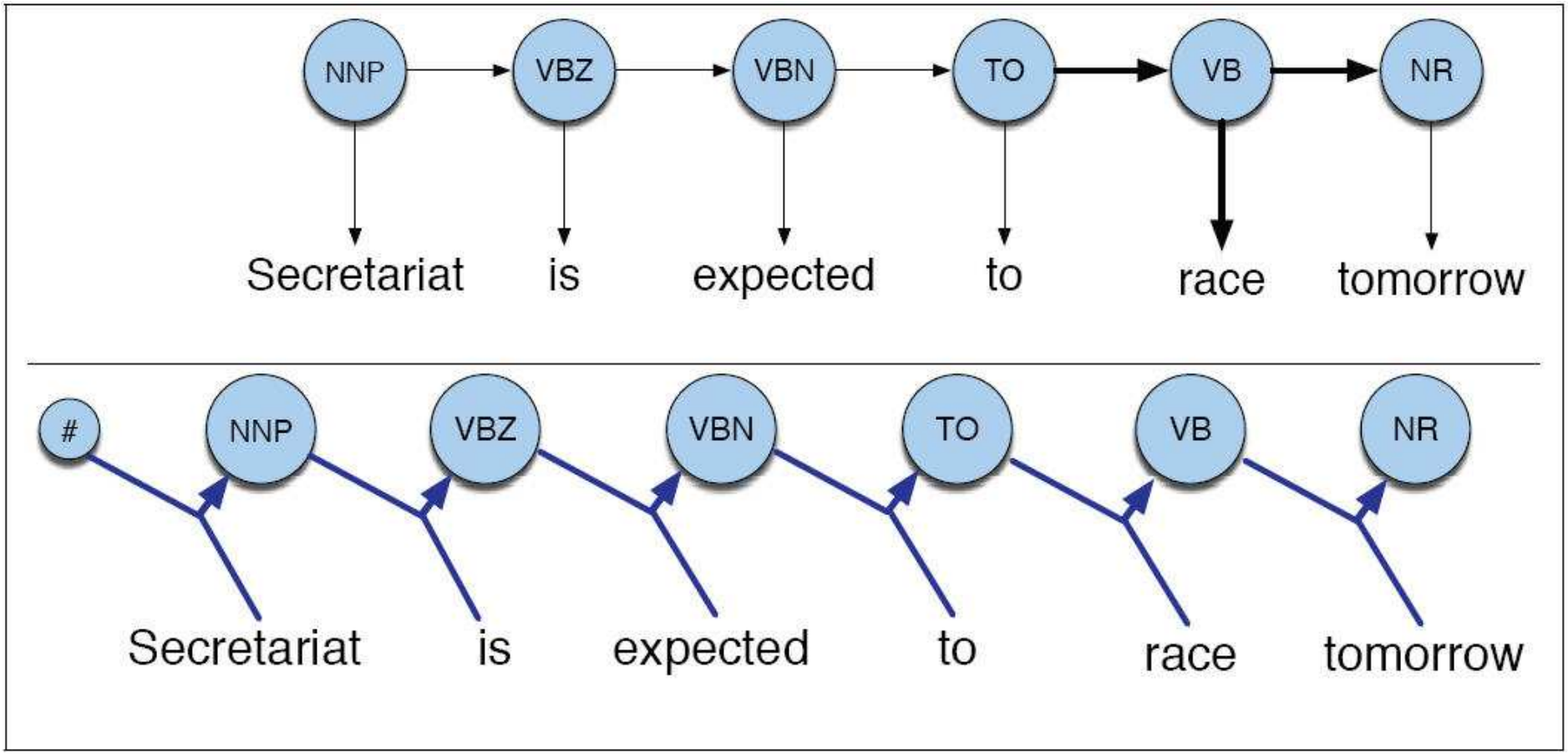
- A Hidden Markov Model (HMM) is a generative model:A

$$\begin{aligned}\hat{T} = & \arg\max_T P(T \mid W) \\ & \arg\max_T P(W \mid T)P(T) \\ & \arg\max_T \prod_i P(w_i \mid t_i) \prod_i (t_i \mid t_{i-1})\end{aligned}$$

- A MaxEnt Markov Model (MEMM) is a discriminative model:

$$\begin{aligned}\hat{T} = & \arg\max_T P(T \mid W) \\ & \arg\max_T \prod_i (t_i \mid w_i, t_{i-1})\end{aligned}$$

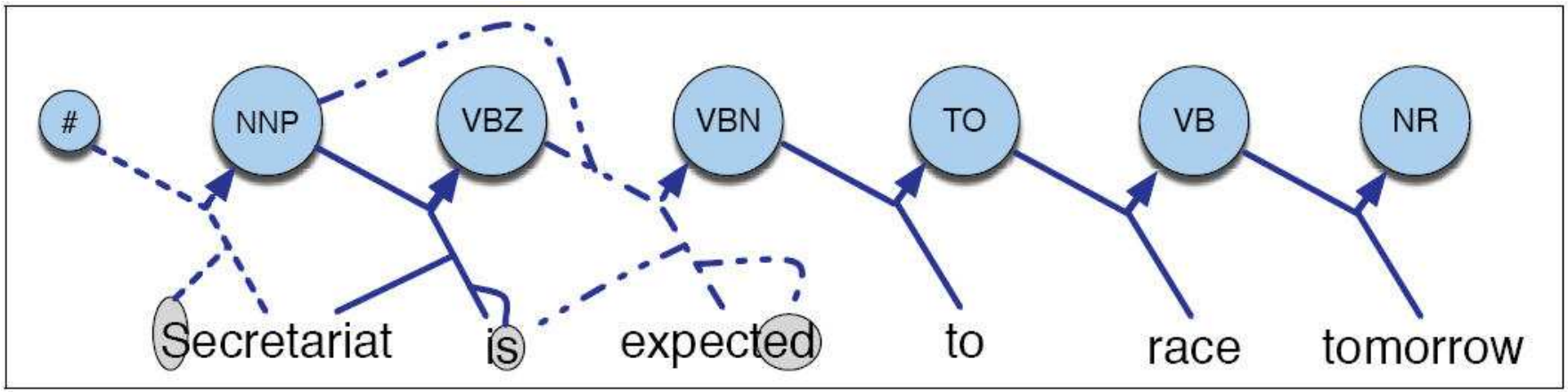
HMM


NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow
#
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow

MEMM

Advantages of MEMM

- HMMs handle emission probabilities and transition probabilities
- The MEMM can incorporate probabilities on the features that we want:



so the probability of transiting a state q to a state q' that produces the observation o is defined as:

$$P(q \mid q', o) = \frac{1}{Z(q', 0)} \exp\left(\sum_i w_i f_i(q, o)\right)$$

Execution of a MEMM

- Viterbi for HMM:

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i)\, a_{ij}\, b_j(o_{t+1}) \quad t = 1, 2, \ldots, T-1, \quad 1 \le j \le N$$

$$= \max_{1 \le i \le N} \delta_t(i)\, P(t_j \mid t_i) P(o_{t+1} \mid t_j) \quad t = 1, 2, \ldots, T-1, \quad 1 \le j \le N$$

- Viterbi for MEMM:

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i)\, P(t_j \mid t_i, o_{t+1}) \quad t = 1, 2, \ldots, T-1, \quad 1 \le j \le N$$

Outline

Why do they call it love when they mean sex?

Or put another way **why do they call it maximum entropy when they mean multinomial logistic regression?**

Where's entropy?

$$H(x) = -\sum_x P(x) \log_2 P(x)$$

Because...

- The intuition of MaxEnt is that the probabilistic model must follow the restrictions we impose, but it should not assume anything special about everything else (i.e. it should leave everything else with the maximum possible entropy).
- Formally:

  Select a model from a set of allowed probability distributions, choose the model $p^* \in \mathcal{C}$ *with maximum entropy* $H(p)$*:*

  $$p^* = \arg\max_{p \in \mathcal{C}} H(p)$$

The solution to this problem is the probability distribution of a multinomial logistic regression model whose weights maximize the likelihood of the training data

End