Maximum entropy and MEMMs

based on Ch. 6 of Jurafsky & Martin

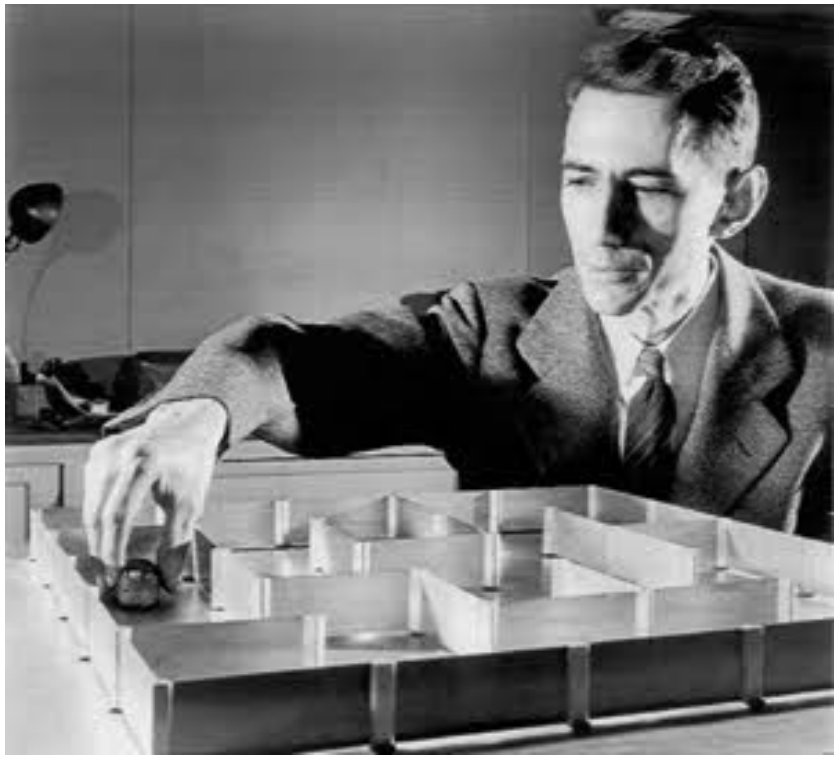
Grzegorz Chrupała

National Centre for Language Technology
School of Computing
Dublin City University

NCLT Seminar 2007

Outline

Markov and Shannon



Outline

Supervised learning

In supervised learning we try to learn a function $h : \mathcal{X} \to \mathcal{Y}$ where $x \in \mathcal{X}$ are inputs and $y \in \mathcal{Y}$ are outputs.

- Binary classification: $\mathcal{Y} = \{-1, +1\}$
- Multiclass classification: $\mathcal{Y} = \{1, \ldots, K\}$ (finite set of labels)
- Regression: $\mathcal{Y} = \mathbb{R}$

The prediction is based on the *feature function* $\Phi : \mathcal{X} \to \mathcal{F}$ where usually $\mathcal{F} = \mathbb{R}^D$ (D-dimensional vector space)

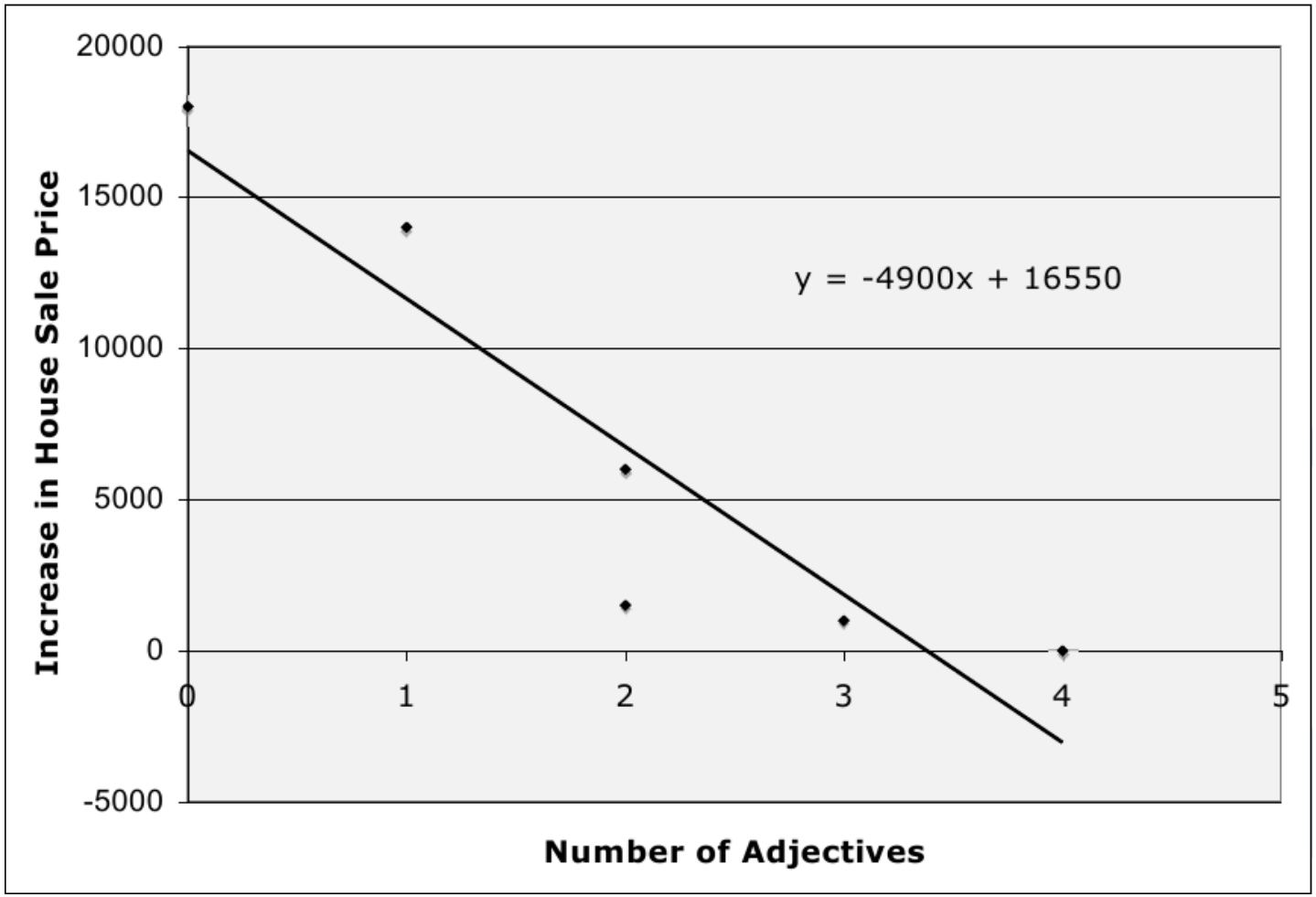
Sequence labeling

- In sequence labeling the function h to learn is instanciated as $h : \mathcal{W}^n \rightarrow \mathcal{L}^n$ where $w \in \mathcal{W}$ are elements of the sequence and $l \in \mathcal{L}$ are labels.
- Input and output sequences are of equal length
- When this is not the case (i.e. chunking, NER) BIO-encoding can be used
- Maximum Entropy Markov Models combine predictions of a MaxEnt classifier using the Viterbi algorithm to find the best sequence of labels

Outline

Linear Regression

- Training data: observations paired with outcomes ($n \in \mathbb{R}$)
- Observations have features (predictors, typically also real numbers)
- The model is a regression line $y = ax + b$ which best fits the observations
  - a is the slope
  - b is the intercept
  - This model has two parameters (or weigths)
  - One feature $= x$
  - Example:
    - $x =$ number of vague adjectives in property descriptions
    - $y =$ amount house sold over asking price


20000
15000
10000
5000
0
-5000
Increase in House Sale Price
y = -4900x + 16550
0
1
2
3
4
5
Number of Adjectives

Multiple linear regression

- More generally $y = w_0 + \sum_{i=0}^{N} w_i f_i$, where
  - y = outcome
  - w_0 = intercept
  - $f_1..f_N$ = features vector and $w_1..w_N$ weight vector
- We ignore w_0 by adding a special f_0 feature, then the equation is equivalent to dot product: $y = \mathbf{w} \cdot \mathbf{f}$

Learning linear regression

- Minimize sum squared error over the training set of M examples

$$cost(W) = \sum_{j=0}^{M} (y_{pred}^{(j)} - y_{obs}^{(j)})^2$$

where

$$y_{pred}^{j} = \sum_{i=0}^{N} w_i f_i^{(j)}$$

- Closed-form formula for choosing the best set of weights W is given by:

$$W = (X^T X)^{-1} X^T \overrightarrow{y}$$

where the matrix X contains training example features, and $\overrightarrow{y}$ is the vector of outcomes.

Logistic regression

- In logistic regression we use the linear model to do classification, i.e. assign probabilities to class labels
- For binary classification, predict $p(y = true|x)$. But predictions of linear regression model are $\in \mathbb{R}$, whereas $p(y = true|x) \in [0, 1]$
- Instead predict logit function of the probability:

$$ln\left(\frac{p(y = true|x)}{1 - p(y = true|x)}\right) = \mathbf{w} \cdot \mathbf{f} \tag{1}$$

$$\frac{p(y = true|x)}{1 - p(y = true|x)} = e^{\mathbf{w}\cdot\mathbf{f}} \tag{2}$$

- Solving for $p(y = true|x)$ we obtain:

$$p(y = true|x) = \frac{e^{\mathbf{w}\cdot\mathbf{f}}}{1 + e^{\mathbf{w}\cdot\mathbf{f}}} \tag{3}$$

$$= \frac{exp\left(\sum_{i=0}^{N} w_i f_i\right)}{1 + exp\left(\sum_{i=0}^{N} w_i f_i\right)} \tag{4}$$

Logistic regression - classification

- Example x belongs to class *true* if:

$$\frac{p(y = true|x)}{1 - p(y = true|x)} > 1 \tag{5}$$

$$e^{\mathbf{w}\cdot\mathbf{f}} > 1 \tag{6}$$

$$\mathbf{w} \cdot \mathbf{f} > 0 \tag{7}$$

$$\sum_{i=0}^{N} w_i f_i > 0 \tag{8}$$

- The equation $\sum_{i=0}^{N} w_i f_i = 0$ defines the hyperplane in N-dimensional space, with points above this hyperplane belonging to class *true*

Logistic regression - learning

- Conditional likelihood estimation: choose the weights which make the probability of the observed values y be the highest, given the observations x
- For the training set with M examples:

$$\hat{\mathbf{w}} = \underset{w}{\operatorname{argmax}} \prod_{i=0}^{M} P(y^{(i)}|x^{(i)})$$

- A problem in convex optimization (not covered here)
  - L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno method)
  - gradient ascent
  - conjugate gradient
  - iterative scaling algorithms

DCU

Outline

Maximum Entropy model

- Logistic regression with more than two classes = multinomial logistic regression
- Also known as Maximum Entropy (MaxEnt)
- The MaxEnt equation generalizes (4) above:

$$p(c|x) = \frac{exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)}{\sum_{c' \in C} exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)} \tag{9}$$

- The denominator is the normalization factor usually called Z used to make the score into a proper probability distribution

$$p(c|x) = \frac{1}{Z} exp \sum_{i=0}^{N} w_{ci} f_i$$

MaxEnt features

- In Maxent modeling normally binary freatures are used
- Features depend on classes: $f_i(c, x) \in \{0, 1\}$
- Those are indicator features
- Example x:
  Secretariat/NNP is/BEZ expected/VBN to/TO race/VB tomorrow
- Example features:

$$f_1(c, x) = \begin{cases} 1 \text{ if } word_i = race \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{cases}$$

$$f_2(c, x) = \begin{cases} 1 \text{ if } t_{i-1} = TO \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{cases}$$

$$f_3(c, x) = \begin{cases} 1 \text{ if } suffix(word_i) = ing \ \& \ c = \text{VBG} \\ 0 \text{ otherwise} \end{cases}$$

Binarizing features

- Example x:
  Secretariat/NNP is/BEZ expected/VBN to/TO race/VB tomorrow
- Vector of symbolic features of x:

$word_i$	suf	tag_{i-1}	is-case(w_i)
race	ace	TO	TRUE

- Class-dependent indicator features of x:

	$word_i$=race	suf=ing	suf=ace	tag_{i-1}=TO	tag_{i-1}=DT	is-lower(w_i)=TRUE	...
JJ	0	0	0	0	0	0	
VB	1	0	1	1	0	1	
NN	0	0	0	0	0	0	
...							

Complex features

- Consider proper nouns: a word is likely to be a proper noun if it starts with a capital letter **and is not** the first word of the sentence.
- We need a conjunction of two primitive features

$$f_{123}(c, x) = \begin{cases} 1 \text{ if } word_{i-1} \neq \langle s \rangle \ \& \ capitalized(w_i) \ \& \ c = \text{NNP} \\ 0 \text{ otherwise} \end{cases}$$

- Digression: some ML methods (e.g. SVMs) use kernels which can capture feature interactions
  - If model can be rewritten in terms of dot products, those can be replaced by kernel functions
  - Using a kernel is equivalent to mapping the input vectors to higher dimensional feature representation
  - E.g. quadratic kernel $K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} \cdot \mathbf{z} \rangle^2$ maps to:

$$\Phi(x) = (x_1x_1, ..., x_1x_n, x_2x_1, ..., x_2x_n, ..., x_nx_1, ..., x_nx_n)$$

- MaxEnt does not do that!

Entia non sunt multiplicanda praeter necessitatem

Entropy

- Out of all possible models, choose the simplest one consistent with the data (Occam's razor)
- Entropy of the distribution of discrete random variable X:

$$H(X) = -\sum_{x} P(X = x) log_2 P(X = x)$$

- The uniform distribution has the highest entropy
- Finding the maximum entropy distribution in the set C of possible distributions

$$p* = \underset{p \in C}{\operatorname{argmax}}\, H(p)$$

- Berger et al. (1996) showed that solving this optimization problem is equivalent to finding the multinomial logistic regression model whose weights maximize the likelihood of the training data.

Outline

HMMs and MEMMs

- HMM POS tagging model:

$$\hat{T} = \operatorname*{argmax}_T P(T|W) \tag{10}$$

$$= \operatorname*{argmax}_T P(W|T)P(T) \tag{11}$$

$$= \operatorname*{argmax}_T \prod_i P(word_i|tag_i) \prod_i P(tag_i|tag_{i-1}) \tag{12}$$

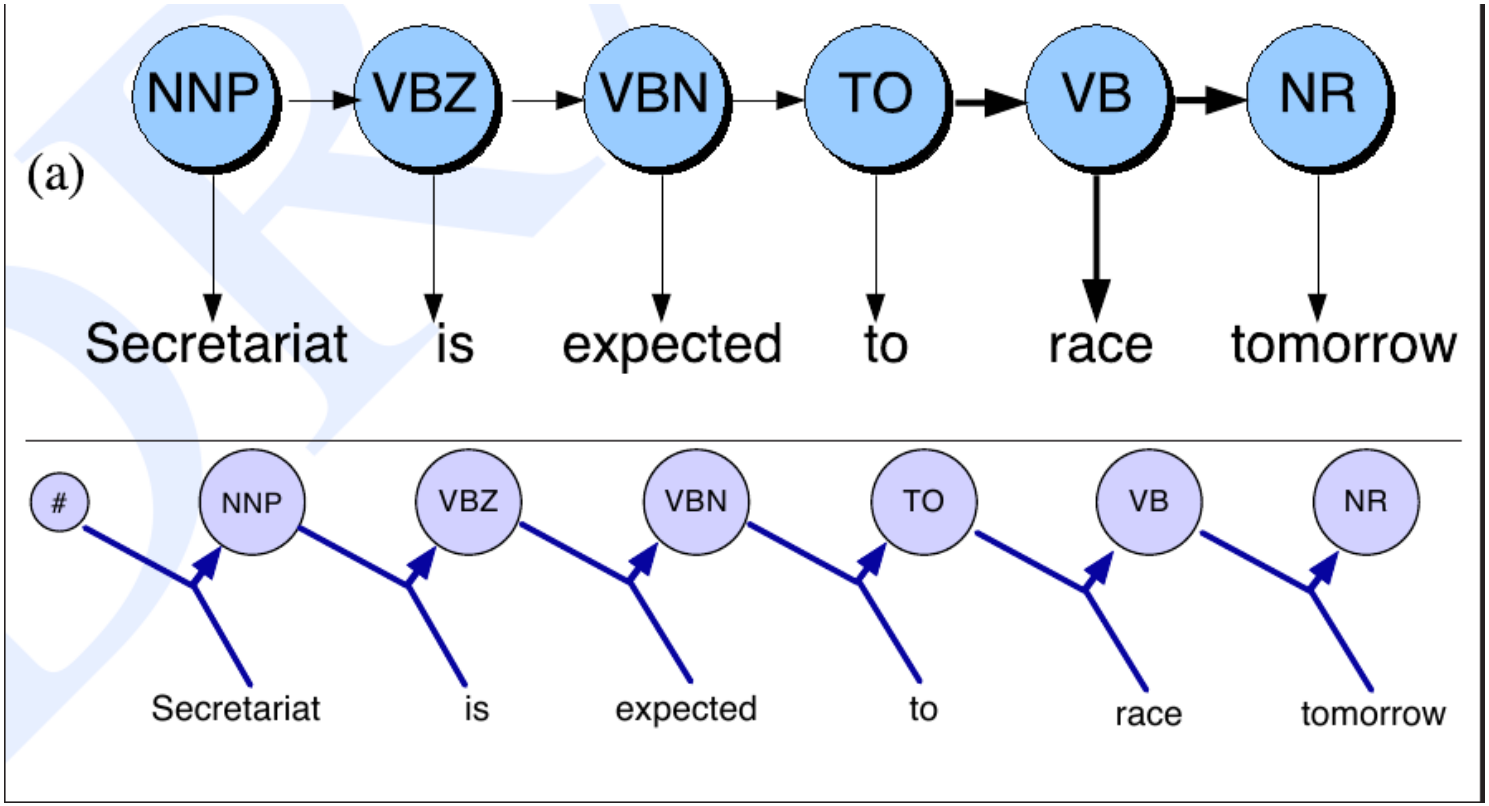
- MEMM POS tagging model:

$$\hat{T} = \operatorname*{argmax}_T P(T|W) \tag{13}$$

$$= \operatorname*{argmax}_T \prod_i P(tag_i|word_i, tag_{i-1}) \tag{14}$$

- Maximum entropy model gives conditional probabilities

Conditioning probabilities in a HMM and a MEMM



Viterbi in MEMMs

- Decoding works almost the same as in HMM
- Except entries in the DP table are values of $P(t_i|t_{i-1}, word_i)$
- Recursive step: Viterbi value of time t for state j:

$$v_t(j) = \max_{i=i}^{N} v_{t-1}(i) P(s_j|s_i, o_t) \quad 1 \leq j \leq N, 1 < t \leq T \tag{15}$$

More

- Accessible intro to MaxEnt: A simple introduction to maximum entropy models for natural language processing, Ratnaparkhi (1997)
- More complete: A maximum entropy approach to natural language processing, Berger et al. (1996) in CL
- MEMM paper: Maximum Entropy Markov Models for Information Extraction and Segmentation, McCallum (2000)
- Other sequence labeling methods
  - Conditional Random Fields (also based on MaxEnt): Lafferty et al (2001)
  - Max-Margin Markov Networks, Taskar et al. (2003)
  - SVM^{struct} Tsochantaridis et al. (2005)

DCU