Lecture 7
Maximum Entropy Models

CS 6320
Outline

- Maximum Entropy Models—Background
- Maximum Entropy Model applied to NLP classification
- Maximum Entropy Markov Models
Maximum Entropy

- Probabilistic machine learning for
  - sequence classification (POS tagging, speech recognition)
  - non-sequential classification (text classification, sentiment analysis)
- Maximum entropy extracts features from inputs, then combines them to classify inputs.
- Computes the probability of a class $c$ given an observation $x$ described by a vector of features $f$

$$p(c | x) = \frac{1}{Z} \exp \sum_i w_i f_i$$
Linear Regression

Problem: Price a house based on “vague” adjectives used in the ads. Ex: fantastic, cute, charming

<table>
<thead>
<tr>
<th># of Vague Adjectives</th>
<th>Amount House Sold Over Asking Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$1000</td>
</tr>
<tr>
<td>2</td>
<td>$1500</td>
</tr>
<tr>
<td>2</td>
<td>$6000</td>
</tr>
<tr>
<td>1</td>
<td>$14000</td>
</tr>
<tr>
<td>0</td>
<td>$18000</td>
</tr>
</tbody>
</table>

Figure 6.17 Some made-up data on the number of vague adjectives (fantastic, cute, charming) in a real estate ad and the amount the house sold for over the asking price.

price = w₀ + w₁ * Num_Adjectives

Figure 6.18 A plot of the (made-up) points in Fig. 6.17 and the regression line that best fits them, with the equation y = -4900x + 16550.
Multiple Linear Regression

In reality, the price of house depends on several factors.

\[
\text{price} = w_0 + w_1 \times \text{Num\_Adjectives} + w_2 \times \text{Mortgage\_Rate} + w_3 \times \text{Num\_Unsold\_Houses}
\]

\[
\text{price} = w_0 + \sum_{i=1}^{N} w_1 \times f_i
\]

Linear regression: \[ y = \sum_{i=0}^{N} w_i \times f_i \]

Dot product: \[ a \cdot b = \sum_{i=1}^{N} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \]

\[ y = w \cdot f \]
Learning in Linear Regression

Problem: Learn the weights $w_i$

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

Minimize the cost function produced by weights $w_i$ for all $M$ examples in the training set.

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

$$W = (X^TX)^{-1}X^T\hat{y}$$
Logistic Regression

- Linear regression predicts real-value functions
- Classification problems deal with discrete values (or classes)
- We calculate the probability that an observation is in a particular class, and pick the class with the highest probability.
- Let observation \( x \) have feature vector \( f_i \), and class \( y \)

\[
P(y = \text{true} \mid x) = \sum_{i=0}^{N} w_i \times f_i = w \cdot f
\]

- Use a model to predict the odds of \( y \) being true

\[
\frac{p(y = \text{true} \mid x)}{1 - p(y = \text{true} \mid x)} = w \cdot f
\]

\[
\ln \left( \frac{p(y = \text{true} \mid x)}{1 - p(y = \text{true} \mid x)} \right) = w \cdot f
\]
Logit Function

\[
\text{logit } (p(x)) = \ln \left( \frac{p(x)}{1 - p(x)} \right)
\]

\[
\ln \left( \frac{p(y = \text{true} | x)}{1 - p(y = \text{true} | x)} \right) = w \cdot f
\]

\[
p(y = \text{true} | x) = e^{w \cdot f}
\]

\[
p(y = \text{true} | x) = (1 - p(y = \text{true} | x))e^{w \cdot f}
\]

\[
p(y = \text{true} | x) = e^{w \cdot f} - p(y = \text{true} | x)e^{w \cdot f}
\]

\[
p(y = \text{true} | x) + p(y = \text{true} | x)e^{w \cdot f} = e^{w \cdot f}
\]

\[
p(y = \text{true} | x)(1 + e^{w \cdot f}) = e^{w \cdot f}
\]

\[
p(y = \text{true} | x) = \frac{e^{w \cdot f}}{1 + e^{w \cdot f}} = \frac{1}{1 + e^{-w \cdot f}}
\]

\[
p(y = \text{false} | x) = \frac{1}{1 + e^{w \cdot f}} = \frac{e^{-w_f}}{1 + e^{-w_f}}
\]

Logistic Regression is the model in which a linear function is used to estimate a logit of probability.

This is called logistic function.
Logistic Regression--Classification

Problem: Given an observation \( x \) decide if it belongs to class “true” or class “false”.

\[
p(y = \text{true} | x) > p(y = \text{false} | x)
\]

\[
\frac{p(y = \text{true} | x)}{p(y = \text{false} | x)} > 1
\]

\[
\frac{p(y = \text{true} | x)}{1 - p(y = \text{true} | x)} > 1
\]

\[
e^{w \cdot f} > 1
\]

\[
w \cdot f > 0
\]

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

\[
\sum_{i=0}^{N} w_i f_i = 0 \text{ is the equation of a hyperplane}
\]
Maximum Entropy Modeling

In NLP we need to classify problems with multiple classes

\[
p(c \mid x) = \frac{1}{Z} \exp \sum_i w_i f_i
\]

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

\[
Z = \sum_c p(c \mid x) = \sum_{c' \in C} \exp \left( \sum_{i=0}^{N} w_{c'i} f_i \right)
\]

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

In MaxEnt instead of indicator functions \( f \), we use \( f(c,x) \), meaning feature for a particular class \( c \) for a given observation \( x \)
Maximum Entropy Modeling

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

\[ f_1(c, x) = \begin{cases} 
1 & \text{if } \text{word}_i = "\text{race}" \& c = \text{NN} \\
0 & \text{otherwise} 
\end{cases} \]

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

\[ f_3(c, x) = \begin{cases} 
1 & \text{if } \text{suffix (word}_i) = "\text{ing}" \& c = \text{VBG} \\
0 & \text{otherwise} 
\end{cases} \]

\[ f_4(c, x) = \begin{cases} 
1 & \text{if } \text{is_lower_case(word}_i) = "\text{race}" \& c = \text{VB} \\
0 & \text{otherwise} 
\end{cases} \]

\[ f_5(c, x) = \begin{cases} 
1 & \text{if } \text{word}_i = "\text{race}" \& c = \text{VB} \\
0 & \text{otherwise} 
\end{cases} \]

\[ f_6(c, x) = \begin{cases} 
1 & \text{if } t_{i-1} = \text{TO} \& c = \text{NN} \\
0 & \text{otherwise} 
\end{cases} \]
Maximum Entropy Modeling

<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>f</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VB</td>
<td>w</td>
<td>.8</td>
<td>.01</td>
<td>.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>f</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>w</td>
<td>.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
P(\text{NN} | x) = \frac{e^{.8} e^{-1.3}}{e^{.8} e^{-1.3} + e^{.8} e^{.01} e^{.1}} = .20
\]

\[
P(\text{VB} | x) = \frac{e^{.8} e^{.01} e^{.1}}{e^{.8} e^{-1.3} + e^{.8} e^{.01} e^{.1}} = .80
\]

\[
\hat{c} = \arg\max_{c \in C} P(c | x)
\]
Why call it Maximum Entropy?

- **Problem:** Assign a tag to the word *zzfish*.

  Without any prior information

<table>
<thead>
<tr>
<th>NN</th>
<th>JJ</th>
<th>NNS</th>
<th>VB</th>
<th>NNP</th>
<th>IN</th>
<th>MD</th>
<th>UH</th>
<th>SYM</th>
<th>VBG</th>
<th>POS</th>
<th>PRP</th>
<th>CC</th>
<th>CD</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
<td>1/45</td>
</tr>
</tbody>
</table>

  Knowing that only four tags are possible

<table>
<thead>
<tr>
<th>NN</th>
<th>JJ</th>
<th>NNS</th>
<th>VB</th>
<th>NNP</th>
<th>IN</th>
<th>MD</th>
<th>UH</th>
<th>SYM</th>
<th>VBG</th>
<th>POS</th>
<th>PRP</th>
<th>CC</th>
<th>CD</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Entropy equation

\[ H(x) = -\sum_x P(x) \log_2 P(x) \]

\[ P(\text{NN}) + P(\text{JJ}) + P(\text{NNS}) + P(\text{VB}) = 1 \]

\[ P(\text{word is zzfish and } t_i = \text{NN or } t_i = \text{NNS}) = \frac{8}{10} \]

\[
\begin{array}{cccccc}
  \text{NN} & \text{JJ} & \text{NNS} & \text{VB} & \text{NNP} & \ldots \\
  \frac{4}{10} & \frac{1}{10} & \frac{4}{10} & \frac{1}{10} & 0 & \ldots \\
\end{array}
\]

\[ p^* = \text{argmax } H(p) \]

The exponential model for multinomial logistic regression also finds the maximum entropy distribution subject to constraints from feature function.
Maximum Entropy Markov Models (MEMM)

\[
\hat{T} = \arg\max_T P(T \mid W)
\]
\[
= \arg\max_T P(W \mid T) P(T)
\]
\[
= \arg\max_T \prod_i P(\text{word}_i \mid \text{tag}_i) \prod_i P(\text{tag}_i \mid \text{tag}_{i-1})
\]

\[
\hat{T} = \arg\max_T P(T \mid W)
\]
\[
= \arg\max_T \prod_i P(\text{tag}_i \mid \text{word}_i, \text{tag}_{i-1})
\]

Advantages of MEMM

1. We estimate directly the probability of each tag giving the previous tag and observed word.

2. We can condition any useful feature of input observation, which was not possible with HMM.
MEMM

Figure 6.20 The HMM (top) and MEMM (bottom) representation of the probability computation for the correct sequence of tags for the Secretariat sentence. Each arc would be associated with a probability; the HMM computes two separate probabilities for the observation likelihood and the prior, while the MEMM computes a single probability function at each state, conditioned on the previous state and current observation.

\[
P(Q \mid O) = \prod_{i=1}^{n} P(o_i \mid q_i) \times \prod_{i=1}^{n} P(q_i \mid q_{i-1})
\]

\[
P(Q \mid O) = \prod_{i=1}^{n} P(q_i \mid q_{i-1}, o_i)
\]
Figure 6.21 An MEMM for part-of-speech tagging, augmenting the description in Fig. 6.20 by showing that an MEMM can condition on many features of the input, such as capitalization, morphology (ending in -s or –ed), as well as earlier words or tags. We have shown some potential additional features for the first three decisions, using different line styles for each class.

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