Chapter 9: Maximum Entropy Methods in Language Processing

Dietrich Klakow
Chapter 16.2
Motivation and Simple Examples
The concept of maximum entropy can be traced back along multiple threads to Biblical times. Only recently however have computers become powerful enough to permit the widescale application of this concept to real world problems in statistical estimation and pattern recognition.

From: „A Maximum Entropy Approach to Natural Language Processing„, by Adam L Berger, Stephen A Della Pietra, Vincent J Della Pietra
Toy example

- Task:
  - Translate German word in to English
  - Possible alternatives:
    - in, at, within, into, to
Estimate the probabilities

Normalisation:

\[ P(\text{in}) + P(\text{at}) + P(\text{within}) + P(\text{into}) + P(\text{to}) = 1 \]

What is the least biased way of determining the probabilities?
Estimate the probabilities

Uniform Distribution:

\[ P(\text{in}) = \frac{1}{5} \]
\[ P(\text{at}) = \frac{1}{5} \]
\[ P(\text{within}) = \frac{1}{5} \]
\[ P(\text{into}) = \frac{1}{5} \]
\[ P(\text{to}) = \frac{1}{5} \]
Estimate the probabilities

Normalisation:

\[ P(\text{in}) + P(\text{at}) + P(\text{within}) + P(\text{into}) + P(\text{to}) = 1 \]

Additional observation

\[ P(\text{in}) + P(\text{at}) = \frac{3}{10} \]

What is the least biased way of determining the probabilities?
Estimate the probabilities

Solution to problem from previous slide:

\[ P(\text{in}) = \frac{3}{20} \]
\[ P(\text{at}) = \frac{3}{20} \]
\[ P(\text{within}) = \frac{7}{30} \]
\[ P(\text{into}) = \frac{7}{30} \]
\[ P(\text{to}) = \frac{7}{30} \]

Why „maximum entropy method“?
Formal Definition of Entropy

$$H(V) = \mathbb{E}[-\log(p(V))]$$

$$= \sum_{w_i \in V} - p(w_i) \log(p(w_i))$$

where $V$ is a set of symbols and $w_i$ the $i$-th symbol
Example

V is set of two symbols \( V = \{a, b\} \)

\[
P(a) = p \\
P(b) = 1 - p
\]

\[
H = -p \log p - (1-p) \log(1-p)
\]

\[
p = 0 \rightarrow H = 0 \\
p = 1 \rightarrow H = 0
\]
Entropy $H = -p \log p - (1-p) \log(1-p)$

Maximum if probabilities for the two symbols are identical
The Maximum Entropy Method

Maximize the entropy because it gives the least prejudiced distribution.

While maximizing, take constraints into account.
Linear Constraints
What are linear constraints good for

- Formalizing our requirements about the final probability distribution
- Taking into account our knowledge derived from a corpus
- Linear, because nonlinear models are more complex
Extend the translation example to include context

Notation:
x: word in the source language
y: word in the target language

Example sentence fragment:
Source language:
   “Er geht *in* die Schule.“
Target language:
   “He goes *to* school.”
Indicator Functions
(feature functions)

- Try to capture essential information from context

\[ f_1(x, y) = \begin{cases} 
1 & \text{if } y = "to" \text{ and } "geht" \text{ precedes } "in" \\
0 & \text{otherwise} 
\end{cases} \]

\[ f_2(x, y) = \begin{cases} 
1 & \text{if } y = "to" \text{ and } "die Schule" \text{ follows } "in" \\
0 & \text{otherwise} 
\end{cases} \]
Integrate Constraints into Probabilities

- Empirical expectation value of feature

\[ \tilde{p}(f_i) \equiv \sum_{x,y} \tilde{p}(x, y) f_i(x, y) \]

With \( \tilde{p}(x, y) \): empirical distribution on corpus (e.g. relative frequencies)

- Expected value of feature derived from unknown model \( p(y|x) \)

\[ p(f_i) \equiv \sum_{x,y} \tilde{p}(x) p(y \mid x) f_i(x, y) \]
Integrate Constraints into Probabilities

- Requirement: match model to corpus statistics

\[ p(f_i) = \tilde{p}(f_i) \]

\[ \sum_{x,y} \tilde{p}(x) p(y \mid x) f_i(x, y) = \sum_{x,y} \tilde{p}(y, x) f_i(x, y) \]

Linear constraint
Set of possible probability distributions

- All possible probability distributions satisfying constraints:

\[ C \equiv \{ p \in P \mid p(f_i) = \tilde{p}(f_i) \text{ for } i = 1..n \} \]

P: space of all probability distributions

Question: how does P look like for a probability space consisting of 2 symbols (e.g. the two sides of a coin)
Question: how does $P$ look like for a probability space consisting of 2 symbols (e.g. two sides of a coin)
Question: how does $P$ look like for a probability space consisting of 3 symbols (e.g. two sides of a coin)
Examples of Simplex and Constraints
Least biased solution on $C$

- Entropy:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x) p(y \mid x) \log p(y \mid x)$$

- Maximize entropy

$$p_\ast = \arg \max_{p \in C} H(p)$$
Example of linear constraints: a trigram language model

Sequence of words: 
\( u, v, w \)

Desired probability:
\( p(w|u,v) \)

or alternatively
\( p(u,v,w) \)
Linear Constraint

w now plays the role of y
The pair u, v plays the role of x

Example feature function:

\[ f_{w_k}(x, y) = \begin{cases} 
1 & \text{if } y = w_k \\
0 & \text{otherwise}
\end{cases} \]
Resulting Constraint Equation

General constraint equation

\[ \sum_{x,y} \tilde{p}(x)p(y \mid x)f_{w_k}(x, y) = \sum_{x,y} \tilde{p}(y, x)f_{w_k}(x, y) \]

Resulting specific constraint equation

\[ \sum_{u,v} \tilde{p}(u, v)p(w_k \mid u, v) = \tilde{p}(w_k) \]

Similarly for \( \tilde{p}(u_l) \) and \( \tilde{p}(v_m) \)
Bigram Constraint Equation

Feature function

\[ f_{u_l w_k}(x, y) = \begin{cases} 1 & \text{if } y = w_k \text{ and } u_l \text{ is directly preceding } w_k \text{ in } x \\ 0 & \text{otherwise} \end{cases} \]

Constraint equation

\[ \sum_v \tilde{p}(u_l, v) p(w_k \mid u_l, v) = \tilde{p}(u_l w_k) \]

Similarly for \( \tilde{p}(v_m u_l) \) and \( \tilde{p}(v_m * w_k) \)
### Effective Trigram via Log-Linear Interpolation: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>317.7</td>
</tr>
<tr>
<td>Linear combination of bigram constraints</td>
<td>302.1</td>
</tr>
<tr>
<td>Maximum entropy model (bigram constraints only)</td>
<td>250.1</td>
</tr>
<tr>
<td>Trigram</td>
<td>198.4</td>
</tr>
</tbody>
</table>
Training Maximum Entropy Models
Log linear models

• General solution of ME problem:

\[ p_\lambda(y \mid x) = \frac{1}{Z_\lambda(x)} \exp\left( \sum_i \lambda_i f_i(x, y) \right) \]

with

\( \lambda \): parameters still to be determined

\( Z_\lambda(h) \): normalization (calculation costly!!!)
Generalized Iterative Scaling

\[ \lambda_{i+1}^j = \lambda_i^j + \log \left( \frac{\sum_{x,y} \tilde{p}(x, y) f_i(x, y)}{\sum_{x,y} \tilde{p}(x) p_j(y|x) f_i(x, y)} \right)^{e_i} \]

- \( e_i \): scaling of constraint
- A few iterations are sufficient
- Takes quite a lot of CPU time
Alternative Training Schemes

- Improved iterative scaling
- Conjugate gradient
- Fast iterative scaling
- ...

Convergence in a Text Classification Task

GIS: not shown on this graph because it has been shown in older publications that IIS is faster.
Selecting Feature Set

- Measure change in likelihood when adding a feature
- Slow and expensive process
- No standard solution yet
Other Applications of Max.-Ent. Models
Translation from a French sentence $F$ to an English sentence $E$

\[
P(F, A \mid E) = \prod_{i=1}^{\|E\|} p(n(e_i) \mid e_i) \prod_{j=1}^{\|F\|} p(f_i \mid e_{a_j}) p(A \mid E, F)
\]

with

- $p(n \mid e)$: number of French words generated from English word $e$
- $p(f \mid e)$: probability that French word $f$ is generated by $e$
- $p(A \mid E, F)$: probability of particular word order
Text-Classification on Reuters Task

**Features**

<table>
<thead>
<tr>
<th>Word $w^i$</th>
<th>Feature weight $\alpha_i$</th>
<th>$\log_e \alpha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs</td>
<td>2.696</td>
<td>0.992</td>
</tr>
<tr>
<td>mln</td>
<td>1.079</td>
<td>0.076</td>
</tr>
<tr>
<td>cts</td>
<td>12.303</td>
<td>2.510</td>
</tr>
<tr>
<td>;</td>
<td>0.448</td>
<td>-0.803</td>
</tr>
<tr>
<td>&amp;</td>
<td>0.450</td>
<td>-0.798</td>
</tr>
<tr>
<td>000</td>
<td>0.756</td>
<td>-0.280</td>
</tr>
<tr>
<td>loss</td>
<td>4.032</td>
<td>1.394</td>
</tr>
<tr>
<td>'</td>
<td>0.993</td>
<td>-0.007</td>
</tr>
<tr>
<td>&quot;</td>
<td>1.502</td>
<td>0.407</td>
</tr>
<tr>
<td>3</td>
<td>0.435</td>
<td>-0.832</td>
</tr>
<tr>
<td>profit</td>
<td>9.701</td>
<td>2.272</td>
</tr>
<tr>
<td>dlr</td>
<td>0.678</td>
<td>-0.388</td>
</tr>
<tr>
<td>1</td>
<td>1.193</td>
<td>0.177</td>
</tr>
<tr>
<td>pct</td>
<td>0.590</td>
<td>-0.528</td>
</tr>
<tr>
<td>is</td>
<td>0.418</td>
<td>-0.871</td>
</tr>
<tr>
<td>s</td>
<td>0.359</td>
<td>-1.025</td>
</tr>
<tr>
<td>that</td>
<td>0.703</td>
<td>-0.352</td>
</tr>
<tr>
<td>net</td>
<td>6.155</td>
<td>1.817</td>
</tr>
<tr>
<td>lt</td>
<td>3.566</td>
<td>1.271</td>
</tr>
<tr>
<td>at</td>
<td>0.490</td>
<td>-0.713</td>
</tr>
<tr>
<td>$f_{K+1}$</td>
<td>0.967</td>
<td>-0.034</td>
</tr>
</tbody>
</table>

**Results**

<table>
<thead>
<tr>
<th>“earnings” assigned?</th>
<th>“earnings” correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES     1014</td>
</tr>
<tr>
<td>NO</td>
<td>NO      73</td>
</tr>
</tbody>
</table>

96.2% accurate
Question Answering

• Features:
  Question word who $\mapsto$ Answer candidate is person
  Question word who $\mapsto$ Answer candidate has two words
  Question word where $\mapsto$ Answer candidate is location
  ...

Named Entity Tagging

See:

Maximum Entropy Models for Named Entity Recognition
O. Bender, F.J. Och, H. Ney
Proceedings of CoNLL-2003
Probabilistic Context Free Grammars

See:

A maximum-entropy-inspired parser
E. Charniak –
Proceedings of NAACL, 2000
Homework

- Read section 16.2 in Manning + Schütze
Summary

• General framework to train probabilities
  • Include constraints (i.e. observations from corpus)
  • Find least biased probability distribution satisfying all constraints

• Warning:
  • CPU-time intensive
  • Picking the right features important for success