Goal of the section today (4/28/2006)

Run through a concrete example of maximum entropy (maxent) models. You should be able to understand these things at the end of the section:

− What are “features”
− What is being adjusted in the training process
− How to compute the objective function that’s being optimized
− How to compute the derivative (used in optimization process)

This mini task is to classify animals to the category of cats, or bears.

\[ c \in C = \{\text{cat, bear}\} \]

We have seen 3 animals. The first animal (d1) is fuzzy. It has claws and it’s small.

\[ d_1 = [\text{fuzzy, claws, small}] \]

We know it’s a cat.

\[ c_1 = \text{cat} \]

The second animal (d2) is fuzzy. It also has claws, but it’s big.

\[ d_2 = [\text{fuzzy, claws, big}] \]

We know it’s a bear.

\[ c_2 = \text{bear} \]

The third animal (d3) we’ve seen has claws, and its size is medium.

\[ d_3 = [\text{claws, medium}] \]

We know it’s a cat.

\[ c_3 = \text{cat} \]

Question:

Here we have 5 characteristics that can be used to describe our data: being fuzzy, have claws, small size, big size, or medium size. And we have 2 classes: cat or bear.

How many (basic) feature functions do we have, and what are they?
Feature Sets:

In this example, we have 10 features:

\[ f_1(c, d) = 1 \text{ if } c \text{ is cat and } d \text{ is fuzzy} \]
\[ f_2(c, d) = 1 \text{ if } c \text{ is bear and } d \text{ is fuzzy} \]
\[ f_3(c, d) = 1 \text{ if } c \text{ is cat and } d \text{ has claws} \]
\[ f_4(c, d) = 1 \text{ if } c \text{ is bear and } d \text{ has claws} \]
\[ f_5(c, d) = 1 \text{ if } c \text{ is cat and } d \text{ is small} \]
\[ f_6(c, d) = 1 \text{ if } c \text{ is bear and } d \text{ is small} \]
\[ f_7(c, d) = 1 \text{ if } c \text{ is cat and } d \text{ is big} \]
\[ f_8(c, d) = 1 \text{ if } c \text{ is bear and } d \text{ is big} \]
\[ f_9(c, d) = 1 \text{ if } c \text{ is cat and } d \text{ is medium} \]
\[ f_{10}(c, d) = 1 \text{ if } c \text{ is bear and } d \text{ is medium} \]

Parameters:

We have 10 \( \lambda_i \)’s, each of them indicates how important each feature is.

**Definition 1**: \( \text{vote(c)} = \sum_{i} \lambda_{i} f_{i}(c,d) \)

In our example…

Suppose we already have a set of \( \lambda_i \)’s. (see the tables below)

For the first animal \( d_1 = \text{[fuzzy, claws, small]} \)

\( \text{vote(cat)} = \sum_{i=1}^{10} \lambda_{i} f_{i}(\text{cat},d_1) = -0.2 \)

<table>
<thead>
<tr>
<th>( \lambda_i )</th>
<th>( \lambda_i f_{i}(\text{cat},d_1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>-0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>-0.3</td>
<td>0</td>
</tr>
<tr>
<td>-0.6</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>0.8</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \text{vote(cat)} = -0.2 \]
The vote for the other class, bear, is:

\[
vote(\text{bear}) = \sum_{i=1}^{10} \lambda_i f_i(\text{bear}, d_1) = 0.2
\]

<table>
<thead>
<tr>
<th>(\lambda_i)</th>
<th>(f_i(\text{bear}, d_1))</th>
<th>(\lambda_i f_i(\text{bear}, d_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.5</td>
<td>1</td>
<td>-0.5</td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.3</td>
<td>1</td>
<td>-0.3</td>
</tr>
<tr>
<td>-0.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
vote(\text{bear}) = 0.2
\]

Definition 2: probabilistic model

\[
P(c \mid d, \lambda) = \frac{\exp(\sum \lambda_i f_i(c, d))}{\sum_c \exp(\sum \lambda_i f_i(c', d))} = \frac{\exp(vote(c))}{\sum_c \exp(vote(c'))}
\]

In our example…

\[
P(\text{cat} \mid d_1, \lambda) = \frac{\exp(vote(\text{cat}))}{\exp(vote(\text{cat})) + \exp(vote(\text{bear}))} = \frac{\exp(-0.2)}{\exp(-0.2) + \exp(0.2)} \approx 0.4013
\]

\[
P(\text{bear} \mid d_1, \lambda) = \frac{\exp(vote(\text{bear}))}{\exp(vote(\text{cat})) + \exp(vote(\text{bear}))} = \frac{\exp(0.2)}{\exp(-0.2) + \exp(0.2)} \approx 0.5987
\]

Interpretation from this example:
Given the set of \(\lambda_i\)’s in the table, and given that we see an animal with the features [fuzzy, claws, small], we’ll conclude the probability of it being a cat is \(0.4013\), being a bear is \(0.5987\). So we’ll say it’s a bear.
If we go back to our first page, we’ll see that this animal is in our training data, and it’s actually a cat, not a bear!
Question: Intuitively, how do we adjust the \(\lambda_i\)’s so that we can correctly predict this example?
What are we optimizing?

When we’re adjusting the λᵢ’s, we’re aiming at maximizing the (conditional) likelihood of our training data.

\[ P(C \mid D, \lambda) = \prod_{(c,d) \in (C,D)} P(c \mid d, \lambda) \]

It’s equivalent to maximizing the log conditional likelihood.

\[ \log P(C \mid D, \lambda) = \sum_{(c,d) \in (C,D)} \log P(c \mid d, \lambda) \]

What’s necessary for doing the optimization?

Give a set of λᵢ’s, calculate

1. **Objective**: the conditional likelihood of the data \( \log P(C \mid D, \lambda) \)

2. **Derivatives**:

\[ \frac{\partial \log P(C \mid D, \lambda)}{\partial \lambda_i} = \text{actual count}(f_i, C) - \text{predicted count}(f_i, \lambda) = \sum_{(c,d) \in (C,D)} f_i(c, d) - \sum_{(c,d) \in (C,D)} \sum_{c'} P(c' \mid d, \lambda) f_i(c', d) \]

A simple intuition here: (in one-dimensional space):

See the excel file for a detailed example of how to compute the value of the objective function and derivatives, and how to adjust λᵢ’s.