Advanced Natural Language Processing
Information Theory and Maximum Likelihood Estimation

Sharon Goldwater

23 September 2014
Recap: basic probability

- Probability distributions (including joint, conditional)
  - outcomes $\rightarrow$ real numbers (0-1, sum to 1)

- Chain rule, conditional independence, Bayes’ rule
  - All of these ubiquitous in NLP
  - Bayes’ rule: can perform *inference* (assuming we know some relevant distributions)
Today

- Entropy and other concepts from Information Theory
- Estimation: where do we get the probability distributions in the first place?
Entropy

• Definition of entropy:

\[ H(X) = \sum_x -p(x) \log_2 p(x) \]

• Intuitively: a measure of uncertainty/disorder

• If we build a probabilistic model, we want that model to have low entropy (low uncertainty)
Entropy Example

One event

\[ p(a) = 1 \quad \text{and} \quad H(X) = -1 \log_2 1 \]

\[ = 0 \]
Entropy Example

2 equally likely events:

\[ p(a) = 0.5 \]
\[ p(b) = 0.5 \]

\[ H(X) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 \]
\[ = - \log_2 0.5 \]
\[ = 1 \]
Entropy Example

4 equally likely events:

\[ p(a) = 0.25 \]
\[ p(b) = 0.25 \]
\[ p(c) = 0.25 \]
\[ p(d) = 0.25 \]

\[ H(X) = - 0.25 \log_2 0.25 - 0.25 \log_2 0.25 \]
\[ - 0.25 \log_2 0.25 - 0.25 \log_2 0.25 \]
\[ = - \log_2 0.25 \]
\[ = 2 \]
Entropy Example

3 equally likely events and one more likely than the others:

\[
p(a) = 0.7 \\
p(b) = 0.1 \\
p(c) = 0.1 \\
p(d) = 0.1
\]

\[
H(X) = -0.7 \log_2 0.7 - 0.1 \log_2 0.1 \\
     - 0.1 \log_2 0.1 - 0.1 \log_2 0.1 \\
     = -0.7 \log_2 0.7 - 0.3 \log_2 0.1 \\
     = -0.7 \times -0.5146 - 0.3 \times -3.3219 \\
     = 0.36020 + 0.99658 \\
     = 1.35678
\]
Entropy Example

3 equally likely events and one much more likely than the others:

\[ H(X) = -0.97 \log_2 0.97 - 0.01 \log_2 0.01 \]
\[ -0.01 \log_2 0.01 - 0.01 \log_2 0.01 \]
\[ = -0.97 \log_2 0.97 - 0.03 \log_2 0.01 \]
\[ = -0.97 \times -0.04394 - 0.03 \times -6.6439 \]
\[ = 0.04262 + 0.19932 \]
\[ = 0.24194 \]
\[ H(X) = 0 \]
\[ H(X) = 1 \]
\[ H(X) = 2 \]
\[ H(X) = 3 \]
\[ H(X) = 1.35678 \]
\[ H(X) = 0.24194 \]
Entropy as y/n questions

How many yes-no questions (bits) do we need to find out the outcome?

- Uniform distribution with $2^n$ outcomes: $n$ q’s.

- Other cases: entropy is the average number of questions per outcome in a (very) long sequence, where questions can consider multiple outcomes at once.
Entropy as encoding sequences

- Assume that we want to encode a sequence of events $X$
- Each event is encoded by a sequence of bits
- For example
  - Coin flip: heads = 0, tails = 1
  - 4 equally likely events: a = 00, b = 01, c = 10, d = 11
  - 3 events, one more likely than others: a = 0, b = 10, c = 11
  - Morse code: e has shorter code than q
- Average number of bits needed to encode $X \geq$ entropy of $X$
The Entropy of English

- Given a number of words in a text, can we guess the next word $p(w_n|w_1, ..., w_{n-1})$?

- Assuming a model with a limited window size

<table>
<thead>
<tr>
<th>Model</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0th order</td>
<td>4.76</td>
</tr>
<tr>
<td>1st order</td>
<td>4.03</td>
</tr>
<tr>
<td>2nd order</td>
<td>2.8</td>
</tr>
<tr>
<td>human, unlimited</td>
<td>1.3</td>
</tr>
</tbody>
</table>
Mutual Information

- A measure of independence between variables
  - How much (on average) does knowing $Y$ reduce $H(X)$?

$$I(X;Y) = H(X) - H(X|Y)$$

- Ex: on avg, how much more certain will I be about $w_i$ if you tell me $w_{i-1}$?
Pointwise Mutual Information

- MI for two particular outcomes (no average)

- Definition:
  \[ I(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \]

- Ex. Consider \( I(\text{San}, \text{Francisco}) \) vs. \( I(\text{and}, \text{a}) \)

- Will discuss more later in course
Noisy channel model

- Concept from Info Theory, used widely in NLP

\[
\begin{align*}
\text{symbol sequence} & \quad \text{noisy/errorful encoding} & \quad \text{output sequence} \\
P(X) & \quad P(Y|X) & \quad P(Y)
\end{align*}
\]
Noisy channel model

- Concept from Info Theory, used widely in NLP

Symbol sequence $P(X) \rightarrow$ noisy/ errorful encoding sequence $P(Y|X) \rightarrow$ output sequence $P(Y)$

<table>
<thead>
<tr>
<th>Application</th>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spelling Correction</td>
<td>true words</td>
<td>typed words</td>
</tr>
<tr>
<td>OCR</td>
<td>true words</td>
<td>image on page</td>
</tr>
<tr>
<td>Speech recognition</td>
<td>true words</td>
<td>acoustic signal</td>
</tr>
<tr>
<td>Machine translation</td>
<td>words in $L_1$</td>
<td>words in $L_2$</td>
</tr>
</tbody>
</table>
Inference in noisy channel model

- Given some $y$, which $x$ produced it?

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- We want $\arg\max_x P(x|y)$
Inference in noisy channel model

• Given some $y$, which $x$ produced it?

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

• We want

$$\arg\max_x P(x|y) = \arg\max_x \frac{P(y|x)P(x)}{P(y)} = \arg\max_x P(y|x)P(x)$$
Inference in noisy channel model

- Given some $y$, which $x$ produced it?

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- We want

$$\arg\max_x P(x|y) = \arg\max_x \frac{P(y|x)P(x)}{P(y)} = \arg\max_x P(y|x)P(x)$$

- $P(X)$: Language model
- $P(Y|X)$: Noise model (or channel/error/acoustic/translation/etc model)
Where do the probabilities come from?

- Previous examples (e.g., dice rolls, coin flips): physical properties of objects

- Usually, probabilities must be *estimated from data*
Estimating a language model

- We want to know $P(\vec{w}) = P(w_1 \ldots w_n)$ for big $n$ (e.g., sentence).

- Impossible!
Estimating a language model

- We want to know $P(\vec{w}) = P(w_1 \ldots w_n)$ for big $n$ (e.g., sentence).

- Impossible!
  - Sparse data: lots of sentences will never have been seen before.
  - Storage: cannot store probabilities for all possible sentences.
Estimating a language model

\[ P(\vec{w}) = P(w_1 \ldots w_n) \]  
\[ = P(w_n|w_{n-1}, w_{n-2}, \ldots w_1)P(w_{n-1}|w_{n-2}, \ldots w_1) \ldots P(w_1) \]  
\[ \approx P(w_n|w_{n-1}, w_{n-2})P(w_{n-1}|w_{n-2}, w_{n-3}) \ldots P(w_1) \]

- (1) By definition
- (2) Using chain rule
- (3) Makes a conditional independence assumption
  - **Markov** assumption: only a finite history matters \((w_i \text{ is cond. indep. of } w_1 \ldots w_{i-3} \text{ given } w_{i-1}, w_{i-2})\). Here, two word history = **trigram** model.
Estimating word probabilities

- Consider only single words (unigrams) for now.

- How to estimate $P(w)$, e.g., $P(\text{the})$?
Estimating word probabilities

• Consider only single words (unigrams) for now.

• How to estimate \( P(w) \), e.g., \( P(\text{the}) \)?

\[
P_{RF}(w) = \frac{C(w)}{N}
\]

where \( C(w) \) is the count of \( w \) in a large corpus, and \( N = \sum w_i C(w_i) \) is the total number of word tokens in the corpus.

• Called the relative frequency estimator.

• Can we justify this mathematically?
Formalizing the estimation problem

- We have a *model of word behavior* (here, unigram model).

- Our model has some parameters $\theta$ (here, probs it assigns to each word; $\theta$ is a vector).

- We observe some data $d$ (here, words in corpus).

- What is the best choice of $\theta$ given $d$?
Formalizing the estimation problem

• What is the best choice of $\theta$ given $d$?

$$P(\theta|d) = \frac{P(d|\theta)P(\theta)}{P(d)}$$

- $P(\theta)$: prior probability of $\theta$
- $P(d|\theta)$: likelihood
- $P(\theta|d)$: posterior probability of $\theta$ given $d$

• As in noisy channel model, maximize $P(\theta|d)$. 
Maximum-likelihood estimation

• Not obvious what prior should be: maybe just uniform?

\[
\arg\max_{\theta} P(d|\theta)P(\theta) = \arg\max_{\theta} P(d|\theta)
\]

• Choose \( \theta \) to maximize the likelihood.
  – the parameters that make the observed data most probable

• This turns out to be just the relative frequency estimator, i.e.,

\[
P_{ML}(w) = P_{RF}(w) = \frac{C(w)}{N}
\]
ML estimates in practice

- ML estimates are easy to compute

\[
P_{ML}(w) = \frac{C(w)}{N}
\]
\[
P_{ML}(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1)}
\]

- However, they do not work well in practice (stay tuned...)

Sharon Goldwater
ANLP Lecture 2
23 September 2014