CMPT-413
Computational Linguistics

Anoop Sarkar
http://www.cs.sfu.ca/~anoop

February 20, 2012
Quick Guide to Probability Theory

Log Probability

Basics of Information Theory
Quick guide to probability theory

- $P(X)$ means probability that $X$ is true
  - $P(\text{baby is a girl}) = 0.5$
    - percentage of total number of babies that are girls
  - $P(\text{baby girl is named Kiki}) = 0.001$
    - percentage of total number of babies that are named Kiki
Joint probability

- $P(X,Y)$ means probability that $X$ and $Y$ are both true
  - $P(\text{baby girl}, \text{blue eyes})$ percentage of total number of babies that are girls and have blue eyes
Conditional probability

- $P(X \mid Y)$ means probability that $X$ is true when we already know that $Y$ is true
  - $P(\text{baby is named Kiki} \mid \text{baby is a girl}) = 0.002$
  - $P(\text{baby is a girl} \mid \text{baby is named Kiki}) = 1$
Conditional and joint probabilities are related:

\[ P(X \mid Y) = \frac{P(X, Y)}{P(Y)} \]

\[ P(\text{baby is named Kiki} \mid \text{baby is a girl}) = \frac{P(\text{baby is a girl, baby is named Kiki})}{P(\text{baby is a girl})} = \frac{0.001}{0.5} = 0.002 \]
Bayes rule

- Conditional probability re-written as likelihood times prior:

\[
P(X \mid Y) = \frac{P(Y \mid X) \times P(X)}{P(Y)}
\]

- \(P(\text{named Kiki} \mid \text{girl}) = \frac{P(\text{girl} \mid \text{named Kiki}) \times P(\text{named Kiki})}{P(\text{girl})} = \frac{1.0 \times 0.001}{0.5} = 0.002\)
Bayes Rule

\[
P(X \mid Y) = \frac{P(X, Y)}{P(Y)} \quad (1)
\]

\[
P(Y \mid X) = \frac{P(Y, X)}{P(X)} \quad (2)
\]

\[
P(X, Y) = P(Y, X) \quad (3)
\]

\[
P(X \mid Y) \times P(Y) = P(Y \mid X) \times P(X) \quad (4)
\]

\[
P(X \mid Y) = \frac{P(Y \mid X) \times P(X)}{P(Y)} \quad (5)
\]

\[
P(X \mid Y) = P(Y \mid X) \times P(X) \quad (6)
\]
Basic Terms

- $P(e)$ – *a priori* probability or just *prior*
- $P(f \mid e)$ – *conditional* probability. The chance of $f$ given $e$
- $P(e, f)$ – *joint* probability. The chance of $e$ and $f$ both happening.
- If $e$ and $f$ are *independent* then we can write
  \[ P(e, f) = P(e) \times P(f) \]
- If $e$ and $f$ are not *independent* then we can write
  \[ P(e, f) = P(e) \times P(f \mid e) \]
  \[ P(e, f) = P(f) \times ? \]
Basic Terms

▶ Addition of integers:

\[ \sum_{i=1}^{n} i = 1 + 2 + 3 + \ldots + n \]

▶ Product of integers:

\[ \prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \ldots \times n \]

▶ Factoring:

\[ \sum_{i=1}^{n} i \times k = k + 2k + 3k + \ldots + nk = k \sum_{i=1}^{n} i \]

▶ Product with constant:

\[ \prod_{i=1}^{n} i \times k = 1k \times 2k \ldots \times nk = k^n \times \prod_{i=1}^{n} i \]
Probability: Axioms

- $P$ measures total probability of a set of events
- $P(\emptyset) = 0$
- $P($all events$) = 1$
- $P(X) \leq P(Y)$ for any $X \subseteq Y$
- $P(X) + P(Y) = P(X \cup Y)$ provided that $X \cap Y = \emptyset$
  - $P($Kiki is a girl$) + P($Kiki is fictional$) = P($Kiki is a fictional girl$),
    provided there are no real girls called Kiki or persons/objects that are fictional Kiki’s.
Probability Axioms

- All events sum to 1:
  \[
  \sum_{e} P(e) = 1
  \]

- Marginal probability \( P(f) \):
  \[
  P(f) = \sum_{e} P(e, f)
  \]

- Conditional probability:
  \[
  \sum_{e} P(e \mid f) = \sum_{e} \frac{P(e, f)}{P(f)} = \frac{1}{P(f)} \sum_{e} P(e, f) = 1
  \]

- Computing \( P(f) \) from axioms:
  \[
  P(f) = \sum_{e} P(e) \times P(f \mid e)
  \]
Probability: The Chain Rule

- $P(a, b, c, d \mid e)$
- We cannot simply remove items from the left of | (verify that it violates the definitions we have given based on sets)
- In this case we can use the chain rule of probability to rescue us
- $P(a, b, c, d \mid e) = P(d \mid e) \cdot P(c \mid d, e) \cdot P(b \mid c, d, e) \cdot P(a \mid b, c, d, e)$
- To see why this is possible, recall that $P(X \mid Y) = \frac{p(X, Y)}{p(Y)}$
  - $p(a, b, c, d, e) = p(d, e) \cdot p(c, d, e) \cdot p(b, c, d, e) \cdot p(a, b, c, d, e)
- Use chain rule and simplify:

  $$P(a, b, c, d \mid e) = P(d \mid e) \cdot P(c \mid d, e) \cdot P(b \mid c, e) \cdot P(a \mid b, e)$$
Probability: The Chain Rule

\[ P(e_1, e_2, \ldots, e_n) = P(e_1) \times P(e_2 \mid e_1) \times P(e_3 \mid e_1, e_2) \ldots \]

\[ P(e_1, e_2, \ldots, e_n) = \prod_{i=1}^{n} P(e_i \mid e_{i-1}, e_{i-2}, \ldots, e_1) \]
What is $y$ in $P(y)$?

- Shorthand for value assigned to a random variable $Y$, e.g. $Y = y$
- $y$ is an element of some implicit event space: $\mathcal{E}$
The marginal probability $P(y)$ can be computed from $P(x, y)$ as follows:

$$P(y) = \sum_{x \in \mathcal{E}} P(x, y)$$

Finding the value that maximizes the probability value:

$$\hat{x} = \arg \max_{x \in \mathcal{E}} P(x)$$
Quick Guide to Probability Theory

Log Probability

Basics of Information Theory
Log Probability Arithmetic

- Practical problem with tiny $P(e)$ numbers: underflow
- One solution is to use log probabilities:

\[
\log(P(e)) = \log(p_1 \times p_2 \times \ldots \times p_n) = \log(p_1) + \log(p_2) + \ldots + \log(p_n)
\]

- Note that:

\[
x = \exp(\log(x))
\]

- Also more efficient: addition instead of multiplication
Log Probability Arithmetic

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\log(p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>0.1</td>
<td>$-3.32$</td>
</tr>
<tr>
<td>0.2</td>
<td>$-2.32$</td>
</tr>
<tr>
<td>0.3</td>
<td>$-1.74$</td>
</tr>
<tr>
<td>0.4</td>
<td>$-1.32$</td>
</tr>
<tr>
<td>0.5</td>
<td>$-1.00$</td>
</tr>
<tr>
<td>0.6</td>
<td>$-0.74$</td>
</tr>
<tr>
<td>0.7</td>
<td>$-0.51$</td>
</tr>
<tr>
<td>0.8</td>
<td>$-0.32$</td>
</tr>
<tr>
<td>0.9</td>
<td>$-0.15$</td>
</tr>
<tr>
<td>1.0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Log Probability Arithmetic

- So: \((0.5 \times 0.5 \times \ldots 0.5) = (0.5)^n\) might get too small but \((-1 - 1 - 1 - 1) = -n\) is manageable

- Another useful fact when writing code (\(\log_2\) is \(\text{log to the base } 2\)):

\[
\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}
\]
Adding probabilities is expensive to compute: 

\[ \logadd(x, y) = \log(\exp(x) + \exp(y)) \]

A more efficient solution, let \( \text{big} \) be a large constant e.g. \( 10^{30} \):

```python
function \logadd(x, y) : # returns \log(\exp(x) + \exp(y))
    if (y - x) > \log(\text{big}) return y
    elsif (x - y) > \log(\text{big}) return x
    else return min(x, y) + \log(\exp(x - \min(x, y)) + \exp(y - \min(x, y)))
endif
```

There is a more efficient way of computing

\[ \log(\exp(x - \min(x, y)) + \exp(y - \min(x, y))) \]
Log Probability Arithmetic

function \( \text{logadd}(x, y) \):
   if \( (y - x) > \log(\text{big}) \) return \( y \)
   elseif \( (x - y) > \log(\text{big}) \) return \( x \)
   elseif \( x \geq y \) return \( x + \log(1 + \exp(y - x)) \)
      # note that \( \max(x, y) = x \) and \( y - x \leq 0 \)
   else return \( y + \log(\exp(x - y) + 1) \)
      # note that \( \max(x, y) = y \) and \( x - y \leq 0 \)
endif

Also, in ANSI C, \texttt{log1p} efficiently computes \( \log(1 + x) \)

http://www.ling.ohio-state.edu/~jansche/src/logadd.c

In Python, \texttt{numpy.logaddexp2(x1,x2)} for base 2
Quick Guide to Probability Theory

Log Probability

Basics of Information Theory
Information theory is the use of probability theory to quantify and measure “information”.

Consider the task of efficiently sending a message. Sender Alice wants to send several messages to Receiver Bob. Alice wants to do this as efficiently as possible.

Let’s say that Alice is sending a message where the entire message is just one character $a$, e.g. $aaaa$. In this case we can save space by simply sending the length of the message and the single character.
Information Theory

- Now let’s say that Alice is sending a completely random signal to Bob. If it is random then we cannot exploit anything in the message to compress it any further.
- The *upper bound* on the number of bits it takes to transmit some infinite set of messages is what is called entropy.
- This formulation of entropy by Claude Shannon was adapted from thermodynamics, converting information into a quantity that can be measured.
- Information theory is built around this notion of message compression as a way to evaluate the amount of information.
Expectation

- For a probability distribution $p$
- **Expectation** with respect to $p$ is a weighted average:

\[
E_p[x] = \frac{x_1 \cdot p_1 + x_2 \cdot p_2 + \ldots + x_n p_n}{p_1 + p_2 + \ldots + p_n} = \sum_{x \in \mathcal{E}} x \cdot p(x)
\]

- Example: for a six-sided die the expectation is:

\[
E_p[x] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} = 3.5
\]
Entropy

- For a probability distribution $p$
- **Entropy** of $p$ is:

$$H(p) = - \sum_{x \in \mathcal{E}} p(x) \cdot \log_2 p(x)$$

- Any base can be used for the log, but base 2 means that entropy is measured in bits.
- What is the *expected* number of bits with respect to $p$:

$$-E_p[\log_2(x)] = H(p)$$

- Entropy answers the question: What is the upper bound on the number of bits needed to transmit messages from event space $\mathcal{E}$, where $p(x)$ defines the probability of observing $x$. 
Alice wants to bet on a horse race. She has to send a message to her bookie Bob to tell him which horse to bet on.

There are 8 horses. One encoding scheme for the messages is to use a number for each horse. So in bits this would be 001, 010, \ldots
(lower bound on message length $= 3$ bits in this encoding scheme)

Can we do better?
If we know how likely we are to bet on each horse, say based on the horse’s probability of winning, then we can do better.

Let $p$ be the probability distribution given in the table above. The entropy of $p$ is $H(p)$.
Entropy

$H(p) =$

$= - \sum_{i=1}^{8} p(i) \log_2 p(i)$

$= - \left( \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{16} \log_2 \frac{1}{16} + 4 \left( \frac{1}{64} \log_2 \frac{1}{64} \right) \right)$

$= - \left( \frac{1}{2} \times -1 + \frac{1}{4} \times -2 + \frac{1}{8} \times -3 + \frac{1}{16} \times -4 + 4 \left( \frac{1}{64} \times -6 \right) \right)$

$= - \left( -\frac{1}{2} - \frac{1}{2} - \frac{3}{8} - \frac{1}{4} - \frac{3}{8} \right)$

$= 2 \text{ bits}$

What is the entropy when the horses are equally likely to win?

$H(\text{uniform distribution}) = -8 \left( \frac{1}{8} \times -3 \right) = 3 \text{ bits}$
Entropy

- e.g., most likely horse gets code 0, next most likely gets 10, and then 110, 1110, ... 
  many possible coding schemes, this is a simple code to illustrate number of bits needed for a large number of messages ...

- Assume there are 320 messages (one for each race): 
  code 0 occurs 160 times, code 10 occurs 80 times, code 110 occurs 40 times, code 1110 occurs 20 times, code 11110 occurs 5 times.

- Total number of bits for all messages: 
  \[160 \times \text{len}(0) + 80 \times \text{len}(10) + 40 \times \text{len}(110) + 20 \times \text{len}(1110) + 5 \times \text{len}(11110)\]

- Number of bits: 
  \[160 \times 1 + 80 \times 2 + 40 \times 3 + 20 \times 4 + 5 \times 5 = 545\]

- Total number of bits per message (per race): 
  \[\frac{545}{320} \approx 1.7 \text{ bits}\] (always less than 2 bits)
Perplexity

- The value $2^{H(p)}$ is called the **perplexity** of a distribution $p$.
- Perplexity is the weighted average number of choices a random variable has to make.
- Choosing between 8 equally likely horses ($H=3$) is $2^3 = 8$.
- Choosing between the biased horses from before ($H=2$) is $2^2 = 4$. 
Relative Entropy

- In real life, we cannot know for sure the exact winning probability for each horse.
- Let’s say $q$ is the estimate and $p$ is the true probability (say we got $q$ by observing previous races with these horses).
- We define the distance between $q$ and $p$ as the relative entropy: written as $D(q\|p)$

\[
D(q\|p) = -\sum_{x \in \mathcal{X}} q(x) \log_2 \frac{p(x)}{q(x)}
\]

- Note that

\[
D(q\|p) = E_{q(x)} \left[ \log_2 \frac{p(x)}{q(x)} \right]
\]

- The relative entropy is also called the Kullback-Leibler divergence.
The **relative entropy** can be written as the sum of two terms:

\[
D(q\|p) = -\sum_{x \in \mathcal{X}} q(x) \log_2 \frac{p(x)}{q(x)} = -\sum_{x} q(x) \log_2 p(x) + \sum_{x} q(x) \log_2 q(x)
\]

We know that \(H(q) = -\sum_x q(x) \log_2 q(x)\).

Similarly define \(H_q(p) = -\sum_x q(x) \log_2 p(x)\).

\[
D(q\|p) = H_q(p) - H(q)
\]

**Relative entropy** \((q, p)\) is equal to **cross entropy** \((q, p)\) minus **entropy** \((q)\).

The term \(H_q(p)\) is called the **cross entropy**.
Cross Entropy and Relative Entropy

- $H_q(p) \geq H(q)$ always.
- $D(q\|p) \geq 0$ always, and $D(q\|p) = 0$ iff $q = p$
- $D(q\|p)$ is not a true distance:
  - It is asymmetric: $D(q\|p) \neq D(p\|q)$,
  - It does not obey the triangle inequality: $D(p\|r) \not\leq D(p\|q) + D(q\|r)$
Conditional Entropy and Mutual Information

- **Entropy** of a random variable $X$:

  \[ H(X) = - \sum_{x \in \mathcal{E}} p(x) \log_2 p(x) \]

- **Conditional Entropy** between two random variables $X$ and $Y$:

  \[ H(X \mid Y) = - \sum_{x, y \in \mathcal{E}} p(x, y) \log_2 p(x \mid y) \]

- **Mutual Information** between two random variables $X$ and $Y$:

  \[ I(X; Y) = D(p(x, y) \parallel p(x)p(y)) = \sum_{x} \sum_{y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \]