

# CMPT-413

## Computational Linguistics

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

February 7, 2013

## Quick Guide to Probability Theory

Log Probability

Basics of Information Theory

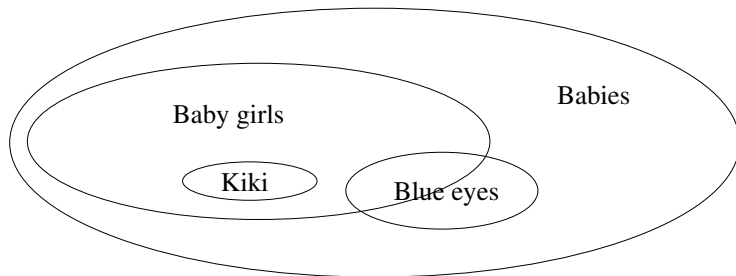
# Quick guide to probability theory

- ▶  $P(X)$  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
  - ▶  $P(\text{baby girl, blue eyes})$  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
  - ▶  $P(\text{baby is named Kiki} \mid \text{baby is a girl}) = 0.002$
  - ▶  $P(\text{baby is a girl} \mid \text{baby is named Kiki}) = 1$

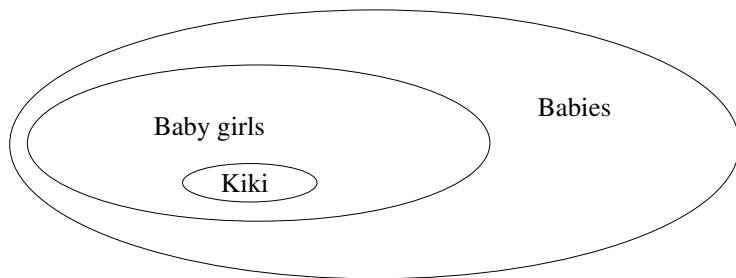


# Conditional probability

- ▶ Conditional and joint probabilities are related:

$$P(X | Y) = \frac{P(X, Y)}{P(Y)}$$

- ▶  $P(\text{baby is named Kiki} | \text{baby is a girl}) = \frac{P(\text{baby is a girl, baby is named Kiki})}{P(\text{baby is a girl})} = \frac{0.001}{0.5} = 0.002$



## Bayes rule

- ▶ Conditional probability re-written as likelihood times prior:

$$P(X | Y) = \frac{P(Y | X) \times P(X)}{P(Y)}$$

- ▶  $P(\text{named Kiki} | \text{girl}) = \frac{P(\text{girl} | \text{named Kiki}) \times P(\text{named Kiki})}{P(\text{girl})} =$   
 $\frac{1.0 \times 0.001}{0.5} = 0.002$



# Bayes Rule

$$P(X | Y) = \frac{P(X, Y)}{P(Y)} \quad (1)$$

$$P(Y | X) = \frac{P(Y, X)}{P(X)} \quad (2)$$

$$P(X, Y) = P(Y, X) \quad (3)$$

$$P(X | Y) \times P(Y) = P(Y | X) \times P(X) \quad (4)$$

$$P(X | Y) = \frac{P(Y | X) \times P(X)}{P(Y)} \quad (5)$$

$$P(X | Y) = P(Y | X) \times P(X) \quad (6)$$



# Basic Terms

- ▶  $P(e)$  – *a priori* probability or just *prior*
- ▶  $P(f | e)$  – *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
$$P(e, f) = P(e) \times P(f | e)$$
$$P(e, f) = P(f) \times ?$$

# Basic Terms

- ▶ Addition of integers:

$$\sum_{i=1}^n i = 1 + 2 + 3 + \dots + n$$

- ▶ Product of integers:

$$\prod_{i=1}^n i = 1 \times 2 \times 3 \times \dots \times n$$

- ▶ Factoring:

$$\sum_{i=1}^n i \times k = k + 2k + 3k + \dots + nk = k \sum_{i=1}^n i$$

- ▶ Product with constant:

$$\prod_{i=1}^n i \times k = 1k \times 2k \dots \times nk = k^n \times \prod_{i=1}^n i$$

# Probability: Axioms

- ▶  $P$  measures total probability of a set of events
- ▶  $P(\emptyset) = 0$
- ▶  $P(\text{all events}) = 1$
- ▶  $P(X) \leq P(Y)$  for any  $X \subseteq Y$
- ▶  $P(X) + P(Y) = P(X \cup Y)$  provided that  $X \cap Y = \emptyset$ 
  - ▶  $P(\text{Kiki is a girl}) + P(\text{Kiki is fictional}) =$   
 $P(\text{Kiki is a fictional girl}),$   
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:

$$\sum_e P(e | f) = \sum_e \frac{P(e, f)}{P(f)} = \frac{1}{P(f)} \sum_e P(e, f) = 1$$

- ▶ Computing  $P(f)$  from axioms:

$$P(f) = \sum_e P(e) \times P(f | e)$$

## 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  $P(X \mid Y) = \frac{p(X, Y)}{p(Y)}$ 
  - ▶  $\frac{p(a, b, c, d, e)}{p(e)} = \frac{p(d, e)}{p(e)} \cdot \frac{p(c, d, e)}{p(d, e)} \cdot \frac{p(b, c, d, e)}{p(c, d, e)} \cdot \frac{p(a, b, c, d, e)}{p(b, c, d, e)}$
- ▶ 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

►  $P(e_1, e_2, \dots, e_n) = P(e_1) \times P(e_2 \mid e_1) \times P(e_3 \mid e_1, e_2) \dots$

$$P(e_1, e_2, \dots, e_n) = \prod_{i=1}^n P(e_i \mid e_{i-1}, e_{i-2}, \dots, e_1)$$

# Probability: Random Variables and Events

- ▶ 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**:  $\mathcal{E}$

# Probability: Random Variables and Events

- ▶ 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:

$$\begin{aligned}\log(P(e)) &= \log(p_1 \times p_2 \times \dots \times p_n) \\ &= \log(p_1) + \log(p_2) + \dots + \log(p_n)\end{aligned}$$

- ▶ Note that:

$$x = \exp(\log(x))$$

- ▶ Also more efficient: addition instead of multiplication

# Log Probability Arithmetic

$p$	$\log(p)$
0.0	$-\infty$
0.1	-3.32
0.2	-2.32
0.3	-1.74
0.4	-1.32
0.5	-1.00
0.6	-0.74
0.7	-0.51
0.8	-0.32
0.9	-0.15
1.0	0.00

# Log Probability Arithmetic

- ▶ So:  $(0.5 \times 0.5 \times \dots 0.5) = (0.5)^n$  might get too small but  $(-1 - 1 - 1 - 1) = -n$  is manageable
- ▶ Another useful fact when writing code ( $\log_2$  is *log to the base 2*):

$$\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$$

# Log Probability Arithmetic

- ▶ Adding probabilities is expensive to compute:  
 $\text{logadd}(x, y) = \log(\exp(x) + \exp(y))$
- ▶ A more efficient soln, let *big* be a large constant e.g.  $10^{30}$ :

```
function logadd(x, y) : # returns  $\log(\exp(x) + \exp(y))$   
if (y - x) > log(big) return y  
elseif (x - y) > log(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 logadd(x, y) :  
    if (y - x) > log(big) return y  
    elif (x - y) > log(big) return x  
    elif (x ≥ y) return x + log(1 + exp(y - x))  
        # note that max(x, y) = x and y - x ≤ 0  
    else return y + log(exp(x - y) + 1)  
        # note that max(x, y) = y and x - y ≤ 0  
    endif
```

Also, in ANSI C, log1p efficiently computes  $\log(1 + x)$

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

In Python, `numpy.logaddexp2(x1,x2)` for base 2

Quick Guide to Probability Theory

Log Probability

Basics of Information Theory

# 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.  $aaaa \dots$ . 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 *expected* 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:

$$\begin{aligned}E_p[x] &= \frac{x_1 \cdot p_1 + x_2 \cdot p_2 + \dots + x_n p_n}{p_1 + p_2 + \dots + p_n} \\&= x_1 \cdot p_1 + x_2 \cdot p_2 + \dots + x_n p_n \\&= \sum_{x \in \mathcal{E}} x \cdot p(x)\end{aligned}$$

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

$$E_p[x] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \dots + 6 \cdot \frac{1}{6} = 3.5$$

# 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 expected number of bits needed to transmit messages from event space  $\mathcal{E}$ , where  $p(x)$  defines the probability of observing  $x$ ?*

# Entropy

- ▶ 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, ...  
(lower bound on message length = 3 bits in this encoding scheme)
- ▶ Can we do better?

# Entropy

Horse 1	$\frac{1}{2}$	Horse 5	$\frac{1}{64}$
Horse 2	$\frac{1}{4}$	Horse 6	$\frac{1}{64}$
Horse 3	$\frac{1}{8}$	Horse 7	$\frac{1}{64}$
Horse 4	$\frac{1}{16}$	Horse 8	$\frac{1}{64}$

- ▶ 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:  $160 \cdot \text{len}(0) + 80 \cdot \text{len}(10) + 40 \cdot \text{len}(110) + 20 \cdot \text{len}(1110) + 5 \cdot \text{len}(11110)$
- ▶ Number of bits:  $160 \cdot 1 + 80 \cdot 2 + 40 \cdot 3 + 20 \cdot 4 + 5 \cdot 5 = 545$
- ▶ Total number of bits per message (per race):  $\frac{545}{320} \approx 1.7$  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  $D(q\|p)$

$$D(q\|p) = - \sum_{x \in \mathcal{E}} q(x) \log_2 \frac{p(x)}{q(x)}$$

- ▶ 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*.

# Cross Entropy and Relative Entropy

- ▶ The **relative entropy** can be written as the sum of two terms:

$$\begin{aligned} D(q\|p) &= - \sum_{x \in \mathcal{E}} q(x) \log_2 \frac{p(x)}{q(x)} \\ &= - \sum_x q(x) \log_2 p(x) + \sum_x q(x) \log_2 q(x) \end{aligned}$$

- ▶ We know that  $H(q) = - \sum_x q(x) \log_2 q(x)$
- ▶ Similarly define  $H_q(p) = - \sum_x q(x) \log_2 p(x)$

$$\begin{aligned} D(q\|p) &= H_q(p) - H(q) \\ \text{relative entropy}(q, p) &= \text{cross entropy}(q, p) - \text{entropy}(q) \end{aligned}$$

- ▶ 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  $Y$ :

$$H(X | Y) = - \sum_{x, y \in \mathcal{E}} p(x, y) \log_2 p(x | y)$$

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

$$I(X; Y) = D(p(x, y) \| p(x)p(y)) = \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}$$