Introduction to Natural Language Processing

a course taught as B4M36NLP at Open Informatics

by members of the Institute of Formal and Applied Linguistics

Today: Week 1, lecture
Today’s topic: Introduction & Probability & Information theory
Today’s teacher: Jan Hajič

E-mail: hajic@ufal.mff.cuni.cz
WWW: http://ufal.mff.cuni.cz/jan-hajic
Intro to NLP

• Instructor: Jan Hajič
  – ÚFAL MFF UK, office: 420 / 422 MS
  – Hours: J. Hajic: Mon 9:00-10:00
  – preferred contact: hajic@ufal.mff.cuni.cz

• Room & time:
  – lecture: Wed, 9:15-10:45
  – seminar [cvičení] follows (Zdenek Zabokrtsky)
  – Final written exam date: Jan 11, 2017
Textbooks you need

• Manning, C. D., Schütze, H.:

• Jurafsky, D., Martin, J.H.:

• Cover, T. M., Thomas, J. A.:

• Jelinek, F.:
Other reading

• Journals:
  – Computational Linguistics
  – Transactions on Computational Linguistics

• Proceedings of major conferences:
  – ACL (Assoc. of Computational Linguistics)
  – EACL (European Chapter of ACL)
  – EMNLP (Empirical Methods in NLP)
  – CoNLL (Natural Language Learning in CL)
  – IJCNLP (Asian chapter of ACL)
  – COLING (Intl. Committee of Computational Linguistics)
Course segments (first three lectures)

• Intro & Probability & Information Theory
  – The very basics: definitions, formulas, examples.

• Language Modeling
  – n-gram models, parameter estimation
  – smoothing (EM algorithm)

• Hidden Markov Models
  – background, algorithms, parameter estimation
Probability
Experiments & Sample Spaces

• Experiment, process, test, ...

• Set of possible basic outcomes: sample space $\Omega$
  - coin toss ($\Omega = \{\text{head, tail}\}$), die ($\Omega = \{1..6\}$)
  - yes/no opinion poll, quality test (bad/good) ($\Omega = \{0,1\}$)
  - lottery ($|\Omega| \approx 10^7$ .. $10^{12}$)
  - # of traffic accidents somewhere per year ($\Omega = \mathbb{N}$)
  - spelling errors ($\Omega = \mathbb{Z}^*$), where $\mathbb{Z}$ is an alphabet, and $\mathbb{Z}^*$ is a set of possible strings over such an alphabet
  - missing word ($|\Omega| \approx$ vocabulary size)
Events

• Event A is a set of basic outcomes
• Usually $A \subseteq \Omega$, and all $A \in 2^\Omega$ (the event space)
  – $\Omega$ is then the certain event, $\emptyset$ is the impossible event
• Example:
  – experiment: three times coin toss
    • $\Omega = \{\text{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}\}$
    – count cases with exactly two tails: then
      • $A = \{\text{HTT, THT, TTH}\}$
    – all heads:
      • $A = \{\text{HHH}\}$
Probability

• Repeat experiment many times, record how many times a given event A occurred (“count” \( c_1 \)).
• Do this whole series many times; remember all \( c_i \)s.
• Observation: if repeated really many times, the ratios of \( c_i/T_i \) (where \( T_i \) is the number of experiments run in the \( i-th \) series) are close to some (unknown but) constant value.
• Call this constant a probability of \( A \). Notation: \( p(A) \)
Estimating probability

• Remember: ... close to an unknown constant.
• We can only estimate it:
  – from a single series (typical case, as mostly the outcome of a series is given to us and we cannot repeat the experiment), set
    \[ p(A) = \frac{c_1}{T_1}. \]
  – otherwise, take the weighted average of all \( \frac{c_i}{T_i} \) (or, if the data allows, simply look at the set of series as if it is a single long series).
• This is the best estimate.
Example

• Recall our example:
  – experiment: three times coin toss
    • $\Omega = \{\text{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}\}$
    – count cases with exactly two tails: $A = \{\text{HTT, THT, TTH}\}$
• Run an experiment 1000 times (i.e. 3000 tosses)
• Counted: 386 cases with two tails ($\text{HTT, THT, or TTH}$)
• estimate: $p(A) = \frac{386}{1000} = .386$
• Run again: 373, 399, 382, 355, 372, 406, 359
  – $p(A) = .379$ (weighted average) or simply $\frac{3032}{8000}$
• Uniform distribution assumption: $p(A) = \frac{3}{8} = .375$
Basic Properties

• Basic properties:
  – p: $2^\Omega \rightarrow [0,1]$
  – $p(\Omega) = 1$
  – Disjoint events: $p(\bigcup A_i) = \sum_i p(A_i)$

• [NB: *axiomatic definition* of probability: take the above three conditions as axioms]

• Immediate consequences:
  – $p(\emptyset) = 0$, $p(\overline{A}) = 1 - p(A)$, $A \subseteq B \Rightarrow p(A) \leq p(B)$
  – $\sum_{a \in \Omega} p(a) = 1$
Joint and Conditional Probability

- $p(A, B) = p(A \cap B)$
- $p(A|B) = \frac{p(A,B)}{p(B)}$

- Estimating form counts:
  - $p(A|B) = \frac{p(A,B)}{p(B)} = \frac{c(A \cap B)}{T} / \frac{c(B)}{T} = \frac{c(A \cap B)}{c(B)}$
Bayes Rule

- $p(A,B) = p(B,A)$ since $p(A \cap B) = p(B \cap A)$

- therefore: $p(A|B) \cdot p(B) = p(B|A) \cdot p(A)$, and therefore

\[
p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}
\]
Independence

• Can we compute $p(A,B)$ from $p(A)$ and $p(B)$?
• Recall from previous foil:
  
  $p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$

  $p(A|B) \cdot p(B) = p(B|A) \cdot p(A)$

  $p(A,B) = p(B|A) \cdot p(A)$

  ... we’re almost there: how $p(B|A)$ relates to $p(B)$?
  
  – $p(B|A) = P(B)$ iff $A$ and $B$ are **independent**

• Example: two coin tosses, weather today and weather on March 4th 1789;

• Any two events for which $p(B|A) = P(B)$!
Chain Rule

\[ p(A_1, A_2, A_3, A_4, ..., A_n) = ! \]

\[ p(A_1|A_2,A_3,A_4,...,A_n) \times p(A_2|A_3,A_4,...,A_n) \times \]
\[ \times p(A_3|A_4,...,A_n) \times ... \times p(A_{n-1}|A_n) \times p(A_n) \]

• this is a direct consequence of the Bayes rule.
The Golden Rule
(of Classic Statistical NLP)

• Interested in an event $A$ given $B$ (when it is not easy or practical or desirable to estimate $p(A|B)$):
• take Bayes rule, max over all $A$s:
• $\arg\max_A p(A|B) = \arg\max_A p(B|A) \cdot p(A) / p(B)$ =

$$\arg\max_A p(B|A) p(A)$$

• ... as $p(B)$ is constant when changing $A$s
Random Variable

- is a function $X: \Omega \to Q$
  - in general: $Q = \mathbb{R}^n$, typically $\mathbb{R}$
  - easier to handle real numbers than real-world events
- random variable is *discrete* if $Q$ is *countable* (i.e. also if finite)
- Example: *die*: natural “numbering” [1,6], *coin*: \{0,1\}
- Probability distribution:
  - $p_X(x) = p(X=x) = \text{df } p(A_x)$ where $A_x = \{a \in \Omega : X(a) = x\}$
  - often just $p(x)$ if it is clear from context what $X$ is
**Expectation**

**Joint and Conditional Distributions**

- is a mean of a random variable (weighted average)
  - \( E(X) = \sum_{x \in X(\Omega)} x \cdot p_X(x) \)

- Example: one six-sided die: 3.5, two dice (sum) 7

- Joint and Conditional distribution rules:
  - analogous to probability of events

- Bayes: \( p_{X|Y}(x,y) = \text{notation } p_{XY}(x|y) = \text{even simpler notation } \)
  \[ p(x|y) = p(y|x) \cdot p(x) / p(y) \]

- Chain rule: \( p(w,x,y,z) = p(z).p(y|z).p(x|y,z).p(w|x,y,z) \)
Essential Information Theory
The Notion of Entropy

• Entropy ~ “chaos”, fuzziness, opposite of order, ...
  – you know it:
    • it is much easier to create “mess” than to tidy things up...

• Comes from physics:
  – Entropy does not go down unless energy is applied

• Measure of *uncertainty*:
  – if low... low uncertainty; the higher the entropy, the higher uncertainty, but the higher “surprise” (information) we can get out of an experiment
The Formula

- Let $p_X(x)$ be a distribution of random variable $X$
- Basic outcomes (alphabet) $\Omega$

$$H(X) = - \sum_{x \in \Omega} p(x) \log_2 p(x)$$

- Unit: bits ($\log_{10}$: nats)
- Notation: $H(X) = H_p(X) = H(p) = H_{X}(p) = H(p_X)$
Using the Formula: Example

• Toss a fair coin: $\Omega = \{\text{head, tail}\}$
  - $p(\text{head}) = .5$, $p(\text{tail}) = .5$
  - $H(p) = -0.5 \log_2(0.5) + (-0.5 \log_2(0.5)) = 2 \times (-0.5) = -1$

• Take fair, 32-sided die: $p(x) = 1 / 32$ for every side $x$
  - $H(p) = -\sum_{i=1}^{32} p(x_i) \log_2 p(x_i) = -32 (p(x_1) \log_2 p(x_1))$ (since for all $i$, $p(x_i) = p(x_1) = 1/32$)
    = $-32 \times ((1/32) \times (-5)) = 5$ (now you see why it’s called bits?)

• Unfair coin:
  - $p(\text{head}) = .2$ ... $H(p) = .722$; $p(\text{head}) = .01$ ... $H(p) = .081
Example: Book Availability

<table>
<thead>
<tr>
<th>Entropy</th>
<th>H(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad bookstore</td>
<td>0</td>
</tr>
<tr>
<td>good bookstore</td>
<td>0.5</td>
</tr>
<tr>
<td>p(Book Available)</td>
<td>1</td>
</tr>
</tbody>
</table>

UFAL MFF UK @ FEL/Intro to Statistical NLP I/Jan Hajic 2016/7
The Limits

• When $H(p) = 0$?
  – if a result of an experiment is known ahead of time:
  – necessarily:
    \[ \exists x \in \Omega; \ p(x) = 1 \ & \ \forall y \in \Omega; \ y \neq x \ \Rightarrow \ p(y) = 0 \]

• Upper bound?
  – none in general
  – for $|\Omega| = n$: $H(p) \leq \log_2 n$
    • nothing can be more uncertain than the uniform distribution
Perplexity: motivation

• Recall:
  – 2 equiprobable outcomes: $H(p) = 1$ bit
  – 32 equiprobable outcomes: $H(p) = 5$ bits
  – 4.3 billion equiprobable outcomes: $H(p) \approx 32$ bits

• What if the outcomes are not equiprobable?
  – 32 outcomes, 2 equiprobable at .5, rest impossible:
    • $H(p) = 1$ bit
    – Any measure for comparing the entropy (i.e. uncertainty/difficulty of prediction) (also) for random variables with different number of outcomes?
Perplexity

- Perplexity:
  \[ G(p) = 2^H(p) \]

- ... so we are back at 32 (for 32 equip. outcomes), 2 for fair coins, etc.

- it is easier to imagine:
  - NLP example: vocabulary size of a vocabulary with uniform distribution, which is equally hard to predict

- the “wilder” (biased) distribution, the better:
  - lower entropy, lower perplexity
Joint Entropy and Conditional Entropy

- Two random variables: \( X \) (space \( \Omega \)), \( Y \) (\( \Psi \))
- Joint entropy:
  - no big deal: \( ((X,Y) \) considered a single event):
    \[
    H(X,Y) = - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 p(x,y)
    \]
- Conditional entropy:
  \[
  H(Y|X) = - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 p(y|x)
  \]

recall that \( H(X) = E(\log_2(1/p_X(x))) \)
(weighted average: weights are not conditional)
Properties of Entropy I

• Entropy is non-negative:
  – $H(X) \geq 0$
  – proof: (recall: $H(X) = - \sum_{x \in \Omega} p(x) \log_2 p(x)$)
    • $\log(p(x))$ is negative or zero for $x \leq 1$,
    • $p(x)$ is non-negative; their product $p(x)\log(p(x))$ is thus negative;
    • sum of negative numbers is negative;
    • and $-f$ is positive for negative $f$

• Chain rule:
  – $H(X,Y) = H(Y|X) + H(X)$, as well as
  – $H(X,Y) = H(X|Y) + H(Y)$ (since $H(Y,X) = H(X,Y)$)
Properties of Entropy II

- Conditional Entropy is better (than unconditional):
  \[ H(Y|X) \leq H(Y) \]

- \( H(X,Y) \leq H(X) + H(Y) \) (follows from the previous (in)equalities)
  - equality iff \( X,Y \) independent
  - [recall: \( X,Y \) independent iff \( p(X,Y) = p(X)p(Y) \)]

- \( H(p) \) is concave (remember the book availability graph?)
  - concave function \( f \) over an interval \((a,b)\):
    \[ \forall x,y \in (a,b), \forall \lambda \in [0,1]: \]
    \[ f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y) \]
  - function \( f \) is convex if \( -f \) is concave
“Coding” Interpretation of Entropy

• The least (average) number of bits needed to encode a message (string, sequence, series,...) (each element having being a result of a random process with some distribution $p$): $= \text{H}(p)$

• Remember various compressing algorithms?
  – they do well on data with repeating (= easily predictable = low entropy) patterns
  – their results though have high entropy $\Rightarrow$ compressing compressed data does nothing
Coding: Example

• How many bits do we need for ISO Latin 1?
  – ⇒ the trivial answer: 8

• Experience: some chars are more common, some (very) rare:
  • ...so what if we use more bits for the rare, and less bits for the frequent?
    [be careful: want to decode (easily)!]
  • suppose: \( p(\text{‘a’}) = 0.3 \), \( p(\text{‘b’}) = 0.3 \), \( p(\text{‘c’}) = 0.3 \), the rest: \( p(x) \approx 0.0004 \)
  • code: ‘a’ ~ 00, ‘b’ ~ 01, ‘c’ ~ 10, rest: \( 11b_1b_2b_3b_4b_5b_6b_7b_8 \)
  • code acbbécbaac: 0010010111000011111001000010
    \[
    \begin{array}{cccccccc}
      a & c & b & b & é & c & b & a & a & c
    \end{array}
    \]
  • number of bits used: 28 (vs. 80 using “naive” coding)

• code length \( \sim 1 / \text{probability} \); conditional prob OK!
Kullback-Leibler Distance (Relative Entropy)

• Remember:
  – long series of experiments... $c_i/T_i$ oscillates around some number... we can only estimate it... to get a distribution $q$.

• So we get a distribution $q$: (sample space $\Omega$, r.v. $X$)

  the true distribution is, however, $p$. (same $\Omega$, $X$)

  $\Rightarrow$ how big error are we making?

• $D(p||q)$ (the Kullback-Leibler distance):

  $D(p||q) = \sum_{x \in \Omega} p(x) \log_2 (p(x)/q(x)) = E_p \log_2 (p(x)/q(x))$
Comments on Relative Entropy

• Conventions:
  – $0 \log 0 = 0$
  – $p \log (p/0) = \infty$ (for $p > 0$)

• Distance? (less “misleading”: Divergence)
  – not quite:
    • not symmetric: $D(p||q) \neq D(q||p)$
    • does not satisfy the triangle inequality
  – but useful to look at it that way

• $H(p) + D(p||q)$: bits needed for encoding $p$ if $q$ is used
Mutual Information (MI)
in terms of relative entropy

• Random variables $X, Y; p_{X \cap Y}(x, y)$, $p_X(x)$, $p_Y(y)$
• Mutual information (between two random variables $X, Y$):

$$I(X, Y) = D(p(x,y) \parallel p(x)p(y))$$

• $I(X,Y)$ measures how much (our knowledge of) $Y$
  contributes (on average) to easing the prediction of $X$
• or, how $p(x,y)$ deviates from (independent) $p(x)p(y)$
Mutual Information: the Formula

• Rewrite the definition: [recall: \( D(r||s) = \sum_{v \in \Omega} r(v) \log_2 (r(v)/s(v)); \)
substitute \( r(v) = p(x,y), s(v) = p(x)p(y); <v> \sim <x,y> \)]

\[
I(X,Y) = D(p(x,y) || p(x)p(y)) = \\
= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 (p(x,y)/p(x)p(y))
\]

• Measured in bits (what else? :-)

From Mutual Information to Entropy

- by how many bits the knowledge of Y lowers the entropy $H(X)$:

$$I(X,Y) = \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 \left( \frac{p(x,y)}{p(y)p(x)} \right) =$$

...use $p(x,y)/p(y) = p(x|y)$

$$= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 \left( \frac{p(x|y)}{p(x)} \right) =$$

...use $\log(a/b) = \log a - \log b$ ($a \sim p(x|y)$, $b \sim p(x)$), distribute sums

$$= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 p(x|y) - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 p(x) =$$

...use def. of $H(X|Y)$ (left term), and $\sum_{y \in \Psi} p(x,y) = p(x)$ (right term)

$$= -H(X|Y) + \left( -\sum_{x \in \Omega} p(x) \log_2 p(x) \right) =$$

...use def. of $H(X)$ (right term), swap terms

$$= H(X) - H(X|Y) \quad \text{...by symmetry, } = H(Y) - H(Y|X)$$
Properties of MI vs. Entropy

• \( I(X,Y) = H(X) - H(X|Y) \) = number of bits the knowledge of \( Y \) lowers the entropy of \( X \)

\[ = H(Y) - H(Y|X) \] (prev. foil, symmetry)

Recall: \( H(X,Y) = H(X|Y) + H(Y) \) \( \Rightarrow \) \( -H(X|Y) = H(Y) - H(X,Y) \) \( \Rightarrow \)

• \( I(X,Y) = H(X) + H(Y) - H(X,Y) \)

• \( I(X,X) = H(X) \) (since \( H(X|X) = 0 \))

• \( I(X,Y) = I(Y,X) \) (just for completeness)

• \( I(X,Y) \geq 0 \) ... let’s prove that now (as promised).
Other (In)Equalities and Facts

• Log sum inequality: for \( r_i, s_i \geq 0 \)
  \[
  \sum_{i=1..n} (r_i \log(r_i/s_i)) \leq (\sum_{i=1..n} r_i) \log(\sum_{i=1..n} r_i / \sum_{i=1..n} s_i))
  \]

• \( D(p||q) \) is convex [in p,q] (\( \Leftrightarrow \) log sum inequality)

• \( H(p_X) \leq \log_2|\Omega| \), where \( \Omega \) is the sample space of \( p_X \)
  
  Proof: uniform \( u(x) \), same sample space \( \Omega \): \( \sum p(x) \log u(x) = -\log_2|\Omega| \);
  \[
  \log_2|\Omega| - H(X) = -\sum p(x) \log u(x) + \sum p(x) \log p(x) = D(p||u) \geq 0
  \]

• \( H(p) \) is concave [in p]:
  
  Proof: from \( H(X) = \log_2|\Omega| - D(p||u) \), \( D(p||u) \) convex \( \Rightarrow H(x) \) concave
Cross-Entropy

• Typical case: we’ve got series of observations
  \( T = \{t_1, t_2, t_3, t_4, ..., t_n\} \) (numbers, words, ...; \( t_i \in \Omega \));

  estimate (simple):
  \[
  \forall y \in \Omega: \hat{p}(y) = \frac{c(y)}{|T|}, \text{ def. } c(y) = |\{t \in T; t = y\}|
  \]

• ...but the true \( p \) is unknown; every sample is too small!
• Natural question: how well do we do using \( \hat{p} \) [instead of \( p \)]?
• Idea: simulate actual \( p \) by using a different \( T' \)
  (or rather: by using different observation we simulate the insufficiency of \( T \) vs. some other data (“random” difference))
Cross Entropy: The Formula

- $H_{p'}(\hat{p}) = H(p') + D(p'\|p)\hat{p}$

\[
H_{p'}(\hat{p}) = -\sum_{x \in \Omega} p'(x) \log_2 (\hat{p}(x))
\]

- $p'$ is certainly not the true $p$, but we can consider it the “real world” distribution against which we test $\hat{p}$

- note on notation (confusing...): $p/p'$ $\leftrightarrow$ $\hat{p}$, also $H_{T'}(p)$

- (Cross)Perplexity: $G_p(p) = G_{T'}(p) = 2^{H_{p'}(\hat{p})}$
Conditional Cross Entropy

• So far: “unconditional” distribution(s) $p(x)$, $p'(x)$...
• In practice: virtually always conditioning on context
• Interested in: sample space $\Psi$, r.v. $Y$, $y \in \Psi$;
  context: sample space $\Omega$, r.v. $X$, $x \in \Omega$;
  “our” distribution $p(y|x)$, test against $p'(y,x)$,

  which is taken from some independent data:

  $$H_{p'}(p) = - \sum_{y \in \Psi, \ x \in \Omega} p'(y,x) \log_2 p(y|x)$$
Sample Space vs. Data

• In practice, it is often inconvenient to sum over the sample space(s) \( \Psi, \Omega \) (especially for cross entropy!)

• Use the following formula:

\[
H_{p'}(p) = - \sum_{y \in \Psi, x \in \Omega} p'(y,x) \log_2 p(y|x) = - \frac{1}{|T'|} \sum_{i=1..|T'|} \log_2 p(y_i|x_i)
\]

• This is in fact the normalized log probability of the “test” data:

\[
H_{p'}(p) = - \frac{1}{|T'|} \log_2 \prod_{i=1..|T'|} p(y_i|x_i)
\]
Computation Example

• **Ω = {a,b,..,z}, prob. distribution (assumed/estimated from data):**
  \[ p(a) = .25, \ p(b) = .5, \ p(\alpha) = 1/64 \text{ for } \alpha \in \{c..r\}, = 0 \text{ for the rest: } s,t,u,v,w,x,y,z \]

• **Data (test):**  barb  \[ p'(a) = p'(r) = .25, \ p'(b) = .5 \]

• **Sum over Ω:**

  \[
  \begin{array}{ccccccccc}
  \alpha & a & b & c & d & e & f & g & \ldots & p & q & r & s & t & \ldots & z \\
  -p'(\alpha)\log_2 p(\alpha) & .5+.5+0+0+0+0+0+0+0+0+1.5+0+0+0+0+0 = 2.5
  \end{array}
  \]

• **Sum over data:**

  \[
  \begin{array}{cccccc}
  i / s_i & 1/b & 2/a & 3/r & 4/b & 1/|T'| \\
  -\log_2 p(s_i) & 1 & + & 2 & + & 6 & + & 1 & = & 10 & (1/4) & \times & 10 & = & 2.5
  \end{array}
  \]
Cross Entropy: Some Observations

• $H(p)$ <, =, > $H_{p'}(p)$: ALL!

• Previous example:

\[
[p(a) = .25, p(b) = .5, p(\alpha) = 1/64 \text{ for } \alpha \in \{c..r\}, = 0 \text{ for the rest: } s,t,u,v,w,x,y,z] \\
H(p) = 2.5 \text{ bits} = H(p') \text{ (barb)}
\]

• Other data: probable:

\[
(1/8)(6+6+6+1+2+1+6+6) = 4.25 \\
H(p) < 4.25 \text{ bits} = H(p') \text{ (probable)}
\]

• And finally: abba:

\[
(1/4)(2+1+1+2) = 1.5 \\
H(p) > 1.5 \text{ bits} = H(p') \text{ (abba)}
\]

• But what about: baby

\[
-p'(\gamma) \log_2 p(\gamma) = -0.25 \log_2 0 = \infty \quad (??)
\]
Cross Entropy: Usage

• Comparing data??
  – **NO!** (we believe that we test on *real* data!)

• Rather: **comparing distributions** *(vs. real data)*

• Have (got) 2 distributions: p and q (on some $\Omega$, $X$)
  – which is better?
  – better: has lower cross-entropy (perplexity) on real data S

• “Real” data: S

• $H_S(p) = - \frac{1}{|S|} \sum_{i=1..|S|} \log_2 p(y_i|x_i)$

• $H_S(q) = - \frac{1}{|S|} \sum_{i=1..|S|} \log_2 q(y_i|x_i)$
Comparing Distributions

Test data S: probable

• \(p(.)\) from prev. example:  
  \[H_S(p) = 4.25\]
  
  \(p(a) = .25, p(b) = .5, p(\alpha) = 1/64\) for \(\alpha \in \{c..r\}\), = 0 for the rest: s,t,u,v,w,x,y,z

• \(q(.|.)\) (conditional; defined by a table):

| q(.|.) \(\downarrow\) | a | b | e | l | o | p | r | other |
|----------------------|---|---|---|---|---|---|---|--------|
| a                    | 0 | .5| 0 | 0 | 0 | 0 | .125| 0      |
| b                    | 1 | 0 | 0 | 0 | 1 | 1 | .125| 0      |
| e                    | 0 | 0 | 0 | 1 | 0 | 0 | .125| 0      |
| l                    | 0 | .5| 0 | 0 | 0 | 0 | .125| 0      |
| o                    | 0 | 0 | 0 | 0 | 0 | 0 | .125| 1      |
| p                    | 0 | 0 | 0 | 0 | 0 | 0 | .125| 0      |
| r                    | 0 | 0 | 0 | 0 | 0 | 0 | .125| 0      |
| other                | 0 | 0 | 1 | 0 | 0 | 0 | .125| 0      |

\[
\frac{1}{8} \times (\log(p|\text{oth.}) + \log(r|p) + \log(o|r) + \log(b|o) + \log(a|b) + \log(b|a) + \log(l|b) + \log(e|l))
\]

\[
\frac{1}{8} \left( 0 + 3 + 0 + 0 + 0 + 1 + 0 + 1 + 0 \right)
\]

\[H_S(q) = .625\]
Language Modeling
(and the Noisy Channel)
The Noisy Channel

• Prototypical case:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output (noisy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,1,1,1,0,1,0,1,…</td>
<td>0,1,1,0,0,1,1,0,…</td>
</tr>
</tbody>
</table>

The channel (adds noise)

• Model: probability of error (noise):

Example: 
- \( p(0|1) = 0.3 \)
- \( p(1|1) = 0.7 \)
- \( p(1|0) = 0.4 \)
- \( p(0|0) = 0.6 \)

• The Task:

Known: the noisy output; want to know: the input (decoding)
Noisy Channel Applications

• OCR
  - straightforward: text → print (adds noise), scan → image

• Handwriting recognition
  - text → neurons, muscles (“noise”), scan/digitize → image

• Speech recognition (dictation, commands, etc.)
  - text → conversion to acoustic signal (“noise”) → acoustic waves

• Machine Translation
  - text in target language → translation (“noise”) → source language

• Also: Part of Speech Tagging
  - sequence of tags → selection of word forms → text
Noisy Channel: The Golden Rule of ...

OCR, ASR, HR, MT, ...

• Recall:
  \[ p(A|B) = \frac{p(B|A) \ p(A)}{p(B)} \]  (Bayes formula)
  \[ A_{\text{best}} = \arg\max_A p(B|A) \ p(A) \]  (The Golden Rule)

• \( p(B|A) \): the acoustic/image/translation/lexical model
  – application-specific name
  – will explore later

• \( p(A) \): the language model
The Perfect Language Model

• Sequence of word forms [forget about tagging for the moment]
• Notation: $A \sim W = (w_1, w_2, w_3, \ldots, w_d)$
• The big (modeling) question:
  $$p(W) = ?$$
• Well, we know (Bayes/chain rule $\rightarrow$):
  $$p(W) = p(w_1, w_2, w_3, \ldots, w_d) =$$
  $$= p(w_1) \times p(w_2|w_1) \times p(w_3|w_1, w_2) \times \ldots \times p(w_d|w_1, w_2, \ldots, w_{d-1})$$
• Not practical (even short $W \rightarrow$ too many parameters)
Markov Chain

• Unlimited memory (cf. previous foil):
  – for $w_i$, we know all its predecessors $w_1, w_2, w_3, \ldots, w_{i-1}$

• Limited memory:
  – we disregard “too old” predecessors
  – remember only $k$ previous words: $w_{i-k}, w_{i-k+1}, \ldots, w_{i-1}$
  – called “$k$th order Markov approximation”

• + stationary character (no change over time):
  \[ p(W) \approx \prod_{i=1}^{d} p(w_i|w_{i-k}, w_{i-k+1}, \ldots, w_{i-1}), \quad d = |W| \]
n-gram Language Models

• (n-1)th order Markov approximation → n-gram LM:

\[
p(W) = \frac{df}{\prod_{i=1..d} p(w_i|w_{i-n+1},w_{i-n+2},...,w_{i-1})}
\]

• In particular (assume vocabulary |V| = 60k):
  • 0-gram LM: uniform model, \( p(w) = 1/|V| \), 1 parameter
  • 1-gram LM: unigram model, \( p(w) \), \( 6 \times 10^4 \) parameters
  • 2-gram LM: bigram model, \( p(w_i|w_{i-1}) \) \( 3.6 \times 10^9 \) parameters
  • 3-gram LM: trigram model, \( p(w_i|w_{i-2},w_{i-1}) \) \( 2.16 \times 10^{14} \) parameters
Maximum Likelihood Estimate

• MLE: Relative Frequency...
  – ...best predicts the data at hand (the “training data”)
• Trigrams from Training Data T:
  – count sequences of three words in T: $c_3(w_{i-2},w_{i-1},w_i)$
    [NB: notation: just saying that the three words follow each other]
  – count sequences of two words in T: $c_2(w_{i-1},w_i)$:
    • either use $c_2(y,z) = \sum_w c_3(y,z,w)$
    • or count differently at the beginning (& end) of data! $p(w_i|w_{i-2},w_{i-1})$

\[ = \text{est. } \frac{c_3(w_{i-2},w_{i-1},w_i)}{c_2(w_{i-2},w_{i-1})} \]
LM: an Example

• Training data:

\[
\text{<s> <s> He can buy the can of soda.}
\]

– Unigram: \( p_1(\text{He}) = p_1(\text{buy}) = p_1(\text{the}) = p_1(\text{of}) = p_1(\text{soda}) = p_1(.) = .125 \)

\( p_1(\text{can}) = .25 \)

– Bigram: \( p_2(\text{He} | \text{<s>}) = 1, p_2(\text{can} | \text{He}) = 1, p_2(\text{buy} | \text{can}) = .5, \)

\( p_2(\text{of} | \text{can}) = .5, p_2(\text{the} | \text{buy}) = 1, \ldots \)

– Trigram: \( p_3(\text{He} | \text{<s>}, \text{<s>}) = 1, p_3(\text{can} | \text{<s>}, \text{He}) = 1, \)

\( p_3(\text{buy} | \text{He}, \text{can}) = 1, p_3(\text{of} | \text{the}, \text{can}) = 1, \ldots, p_3(.) | \text{of}, \text{soda}) = 1. \)

– Entropy: \( H(p_1) = 2.75, H(p_2) = .25, H(p_3) = 0 \leftarrow \text{Great?!} \)
LM: an Example (The Problem)

- Cross-entropy:
- \( S = \langle s \rangle \langle s \rangle \) It was the greatest buy of all.
- Even \( H_S(p_1) \) fails (= \( H_S(p_2) = H_S(p_3) = \infty \)), because:
  - all unigrams but \( p_1(\text{the}), p_1(\text{buy}), p_1(\text{of}) \) and \( p_1(.) \) are 0.
  - all bigram probabilities are 0.
  - all trigram probabilities are 0.
- We want: to make all (theoretically possible*) probabilities non-zero.

*in fact, all: remember our graph from day 1?