Essential Information Theory I

Pavel Rychlý

PA154 Statistické nástroje pro korpusy, Spring 2014

Source

Introduction to Natural Language Processing (600.465)
Dr. Jan Hajič
CS Dept., Johns Hopkins Univ.
hajic@cs.jhu.edu
www.cs.jhu.edu/~hajic

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 used
- Measure of **uncertainty**:
  - if low . . . low uncertainty

Entropy

The higher the entropy, the higher uncertainty, but the higher “surprise” (information) we can get out of experiment.

The Formula

- Let $p_x(x)$ be a distribution of random variable X
- Basic outcomes (alphabet) Ω

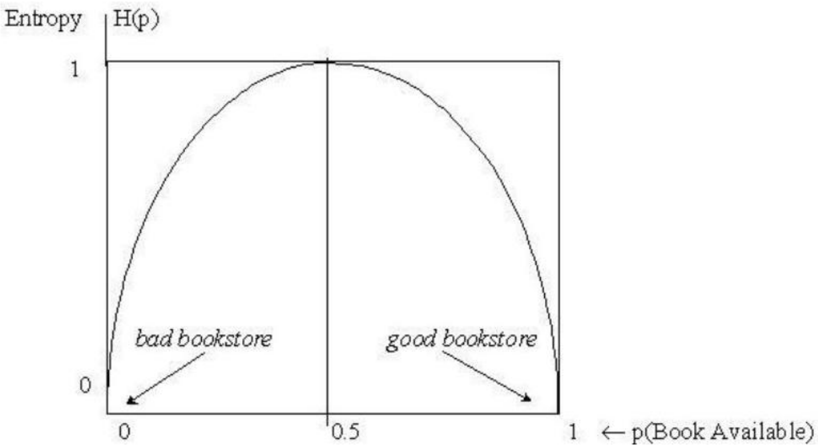
$$\boxed{H(X) = -\sum_{x \in \Omega} p(x) \log_2 p(x)}$$

- 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: $\Omega = \{head, tail\}$
  - p(head) = .5, p(tail) = .5
  - $H(p) = -0.5\log_2(0.5) + (-0.5\log_2(0.5)) = 2 \times ((-0.5) \times (-1)) = 2 \times 0.5 = 1$
- Take fair, 32-sided die: $p(x) = \frac{1}{32}$ for every side x
  - $H(p) = -\sum_{i=1\dots32} p(x_i)\log_2 p(x_i) = -32(p(x_1)\log_2 p(x_1))$ (since for all i $p(x_i) = p(x_1) = \frac{1}{32}$ $= -32 \times (\frac{1}{32} \times (-5)) = 5$ (*now you see why it's called* **bits**?)
- Unfair coin:
  - p(head) = .2 … **H(p) = .722**
  - p(head) = .1 … **H(p) = .081**

Example: Book Availability



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

Entropy and Expectation

- Recall:
  - $E(X) = \sum_{x \in X(\Omega)} p_x(x) \times x$
- Then:
  $E\left(\log_2\left(\frac{1}{p(x)}\right)\right) = \sum_{x \in X(\Omega)} p_x(x) \log_2\left(\frac{1}{p_x(x)}\right) =$
  $-\sum_{x \in X(\Omega)} p_X(x) \log_2 p_x(x) = H(p_x) =_{notation} H(p)$

Perplexity: motivation

- Recall:
  - 2 equiprobable outcomes: $H(p) = 1$ bit
  - 32 equiprobable outcomes: $H(p) = 5$ bits
  - 4.3 billion equiprobable outcomes: $H(p) \cong 32$ bits
- What if the outcomes are not equiprobable?
  - 32 outcomes, 2 equiprobable at 0.5, 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 eqp. 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 Ω), Y (Ψ)
- 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 $H(X) = E\left(\log_2 \frac{1}{p_X(x)}\right)$

(weighted "average", and weights are not conditional)

Conditional Entropy (Using the Calculus)

- other definition:

$$H(Y|X) = \sum_{x \in \Omega} p(x) H(Y|X = x) =$$

for $H(Y|X = x)$, we can use

the single-variable definition (x $\sim$ constant)

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

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

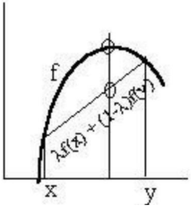
$$= - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x, y) \log_2 p(y|x)$$

Properties of Entropy I

- Entropy is non-negative:
  - $H(X) \geq 0$
  - proof: (recall: $H(X) = -\sum_{x \in \Omega} p(x) \log_2 p(x)$)
    - $\log_2(p(x))$ 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
- for proofs and generalizations, see Cover/Thomas



"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): $= H(p)$
- 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?
  - $\Rightarrow$ 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('a') = 0.3, p('b') = 0.3, p('c') = 0.3, the rest: p(x)$\cong$.0004
  - code: 'a' $\sim$ 00, 'b' $\sim$ 01, 'c' $\sim$ 10, rest: $11 b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$
  - code 'acbbécbaac':

00	10	01	01	1100001111	10	01	00	00	10
a	c	b	b	é	c	b	a	a	c

  - number of bits used: 28 (vs. 80 using "naive" coding)
- code length $\sim \frac{1}{probability}$; conditional prob. OK!

Entropy of Language

- Imagine that we produce the next letter using

$$p(l_{n+1}|l_1, \dots l_n),$$

where $l_1, \dots l_n$ is the sequence of **all** the letters which had been uttered so far (i.e. n is really big!); let's call $l_1, \dots l_n$ the **history** $h(h_{n+1})$, and all histories H:

- Then compute its entropy:
  - $-\sum_{h \in H} \sum_{l \in A} p(l, h) \log_2 p(l|h)$
- Not very practical, isn't it?

Cross-Entropy

- Typical case: we've got series of observations $T = \{t_1, t_2, t_3, t_4, \ldots, t_n\}$ (numbers, words, ...; $t_1 \in \Omega$); estimate (sample): $\forall y \in \Omega : \tilde{p}(y) = \frac{c(y)}{|T|}$, 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 $\tilde{p}$ (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

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

$$\boxed{H_{p'}(\tilde{p}) = -\sum_{x\in\Omega} p'(x)\log_2 \tilde{p}(x)}$$

- p' is certainly not the true p, but we can consider it the "real world" distribution against which we test $\tilde{p}$
- note on notation (confusing ...): $\frac{p}{p'} \leftrightarrow \tilde{p}$, also $H_{T'}(p)$
- (Cross)Perplexity: $G_{p'}(p) = G_{T'}(p) = 2^{H_{p'}(\tilde{p})}$

Conditional Cross Entropy

- So far: "unconditional" distribution(s) $p(x), p'(x)\ldots$
- In practice: virtually always conditioning on context
- Interested in: sample space Ψ, r.v. Y, $y \in \Psi$;
  context: sample space Ω, 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 space(s) Ψ, Ω (especially for cross entropy!)
- Use the following formula:
  $H_{p'}(p) = -\sum_{y\in\Psi, x\in\Omega} p'(y,x) \log_2 p(y|x) =$
  $-1/|T'| \sum_{i=1\ldots|T'|} \log_2 p(y_i|x_i)$
- This is in fact the normalized log probability of the "test" data:

$$H_{p'}(p) = -1/|T'| log_2 \prod_{i=1\ldots|T'|} p(y_i|x_i)$$

Computation Example

- $\Omega = \{a, b, .., z\}$, prob. distribution (assumed/estimated from data):
  p(a) = .25, p(b) = .5, p(α) = $\frac{1}{64}$ for $\alpha \in \{c..r\}$, = 0 for the rest: s,t,u,v,w,x,y,z
- Data (test): <u>barb</u> p'(a) = p'(r) = .25, p'(b) = .5
- Sum over Ω:

  α a b c d e f g ... p q r s t ... z

  $-p'(\alpha)\log_2 p(\alpha)$ .5+.5+0+0+0+0+0+0+0+0+0+1.5+0+0+0+0+0 = <u>2.5</u>

- Sum over data:

  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) × 10 = <u>2.5</u>

Cross Entropy: Some Observations

- $H(p)$??$<, =, >$?? $H_{p'}(p)$: ALL!
- Previous example:

  p(a) = .25, p(b) = .5, p(α)= $\frac{1}{64}$ for $\alpha \in \{c..r\}$, = 0 for the rest: s,t,u,v,w,x,y,z

$$H(p) = 2.5 bits = H(p')(\underline{barb})$$

- Other data: <u>probable</u>: $(\frac{1}{8})(6+6+6+1+2+1+6+6) = 4.25$

$$H(p) < 4.25 bits = H(p')(\underline{probable})$$

- And finally: <u>abba</u>: $(\frac{1}{4})(2+1+1+2) = 1.5$

$$H(p) > 1.5 bits = H(p')(\underline{abba})$$

- But what about: <u>baby</u> $-p'(\text{'}y\text{'}) \log_2 p(\text{'}y\text{'}) = -.25 \log_2 0 = \infty$ (??)

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 Ω, X)
  - which is better?
  - better: has lower cross-entropy (perplexity) on real data S
- "Real" data: S
- $H_S(p) = -1/|S| \sum_{i=1..|S|} log_2 p(y_i|x_i)$??
  $H_S(q) = -1/|S| \sum_{i=1..|S|} log_2 q(y_i|x_i)$

Comparing Distributions

- $p(.)$ from previous example: $H_S(p) = 4.25$

  p(a) = .25, p(b) = .5, p(α) = $\frac{1}{64}$ for $\alpha \in \{c..r\}$, = 0 for the rest: s,t,u,v,w,x,y,z

- $q(.|.)$ (conditional; defined by a table):

q(.|.)→ ↓	a	b	e	l	o	p	r	other
a	0	.5	0	0	0	.125	0	0
b	1	0	0	0	1	.125	0	0
e	0	0	0	1	0	.125	0	0
l	0	.5	0	0	0	.125	0	0
o	0	0	0	0	0	.125	1	0
p	0	0	0	0	0	.125	0	1
r	0	0	0	0	0	.125	0	0
other	0	0	1	0	0	.125	0	0

ex.: q(o|r) = 1

q(r|p) = .125

(1/8) (log(p|oth.)+log(r|p)+log(o|r)+log(b|o)+log(a|b)+log(b|a)+log(l|b)+log(e|l))

(1/8) (0 + 3 + 0 + 0 + 1 + 0 + 1 + 0)

$H_S(q) = .625$