Topics in Natural Language Processing

Shay Cohen

Institute for Language, Cognition and Computation

University of Edinburgh

Lecture 2

Administrativia

Reminder: the requirements for the class are presentations, assignment, brief paper responses and an essay.

- Different topics are available online
- Example topics: topic models, language modeling, parsing, semantics, neural networks (your own topic?)
- Choose whatever level of difficulty you feel comfortable with, so that: (a) your presentation is clear; (b) your brief paper response is informative; (c) the essay goes into details about the topic.

Administrativia

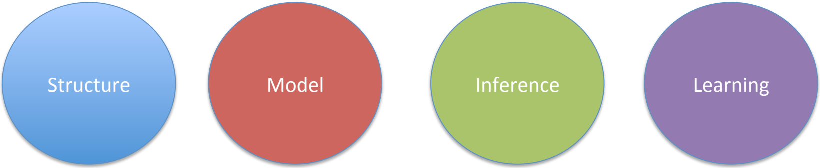
- Presentations start on the week of 12/2
- Please submit the form that I sent by **Friday next week at 4pm (26/1)**
- I will follow-up with an email by some time tomorrow

Today's Class

- Basic refresher about probability
- What is learning?
- What is a statistical model?
- How do we pick a statistical model?

Solving an NLP Problem

When modelling a new problem in NLP, need to address four issues:



Probability and Statistics: Reminder

Probability distribution? Example: unigram model

$\Omega = \{\text{the}, \text{cat}, \text{dog}, \text{sit}, \text{chase}\}$

$p \colon \Omega \to [0, 1]$ - $p(w)$ is the probability attached to w

$p(w) \geq 0$, $\sum_w p(w) = 1$, $\int_w p(w) dw = 1$

Random variables

Random variable:
A function $X\colon \Omega \to \mathbb{R}$

$\Omega = \{\mathrm{the}, \mathrm{dog}, \mathrm{cat}\}$

$X_a(w) =$ count the number of a's in w

$X_a(\mathrm{the}) = 0$, $X_a(\mathrm{cat}) = 1$

$\Omega_2 = \{-\mathrm{ed}, -\mathrm{ing}, -\mathrm{ion}\}$

$X(w) =$ suffix of the word, $X\colon \Omega \to \Omega_2$

Random variables induce probability distributions:

$p(X = \text{ion}) =$ the probability of a word w ending in -ion

$= \sum_{w:\ w \text{ ends in -ion}} p(w)$

$= \sum_w I(w \text{ ends in -ion}) p(w)$

$= E[I(w \text{ ends in ion})]$

where $I(\Gamma)$ is 0 if Γ is false and 1 if Γ is true.

Continuous random variables with density functions:
Guassians for example (mean 0 and standard deviation 1)

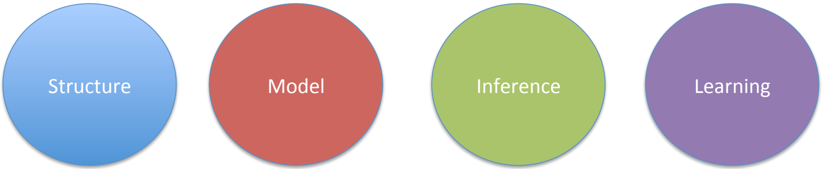
$$p(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

$$p(x \in A) = \int_{x \in A} p(x) dx$$

$$\int_{-\infty}^{\infty} p(x) = 1$$

Solving an NLP Problem

When modelling a new problem in NLP, need to address four issues:



Model Family

A set of probability distributions (unigram example):
$\mathcal{M} = \{p_1, p_2, \ldots\}$

$p_i \colon \Omega \to [0, 1]$

The model family does not have to be countable

Parameters

A set of parameters:
Θ where for each $\theta \in \Theta$ there is $p(w \mid \theta)$

$\mathcal{M} = \{p(w \mid \theta) \mid \theta \in \Theta\}$

$\Omega = \{\text{the}, \text{dog}, \ldots\}$

$p(w) =$ probability of word w
$\Theta \subset \mathbb{R}^{V-1}$ s.t. $0 \leq \theta_i \leq 1$

$\Theta \subset \mathbb{R}^{V}$ s.t. $0 \leq \theta_i \leq 1$ and $\sum_{i=1}^{V} \theta_i = 1$

Estimation

What is training data?

$w^{(1)}, w^{(2)}, w^{(3)}, \ldots \in \Omega$

Estimation

What is training data?

$w^{(1)}, w^{(2)}, w^{(3)}, \ldots \in \Omega$

- True distribution does not have to be a member of the model family
- We usually use the i.i.d. assumption (independently and identically distributed)

Statistical Learning

- What does statistical learning do?
  - Induce a model from data
  - Models tell us how data are generated
  - Learning does the “opposite”

- Two different paradigms to Statistics: frequentist and Bayesian

Approach 1: frequentist Statistics

- We need an objective function $f(\theta, w_1, \ldots, w_n)$
- The higher the value of f is, the better it predicts the training data

$D = \{w_1, \ldots, w_n\}$

$D \rightarrow \Theta$ - that's estimation

$\theta^* = \arg\max_{\theta \in \Theta} f(\theta, w_1, \ldots, w_n)$

In an ideal world...

We have a measure by which we take decisions. Call it ℓ (for loss)

The loss function maps (w, θ) to a number that tells what is the incurred loss if we choose θ for w

If we knew the true distribution, we would choose:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_p[\ell(w, \theta)]$$

Unfortunately we don't have "direct" access to the true distribution (we only have samples). This distribution is exactly what we are trying to model!

We will go back to that...

Choice of f: likelihood

$f(\theta, w_1, \ldots, w_n)$ is a real-valued function

$f(\theta, w_1, \ldots, w_n) = p(w_1, \ldots, w_n \mid \theta) = \prod_{i=1}^{n} p(w_i \theta)$

w_i are independent

Log-likelihood

$f(\theta, w_1, \ldots, w_n) = p(w_1, \ldots, w_n \mid \theta) = \prod_{i=1}^{n} p(w_i \theta)$

$\theta^* = \arg\max_\theta \prod_{i=1}^{n} p(w_i \mid \theta)$ – maximising likelihood

$L(w_1, \ldots, w_n) = \log f(\theta, w_1, \ldots, w_n)$

$\theta^* = \arg\max \log \left(\prod_{i=1}^{n} p(w_i \mid \theta)\right) = \arg\max_\theta \sum_{i=1}^{n} \log p(w_i \mid \theta)$

Next step

Estimation: maximisation of L. The result is the "best" θ that fits to the data *according to the objective function* L

$\theta^* = \arg\max_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \log p(w_i \mid \theta)$

The term maximised is called "average log-likelihood."

Empirical Risk Minimization

We don't have access to the true distribution, but we have samples.

If our $\ell(\theta, w_i) = -\log p(w_i \mid \theta)$ then we are minimizing the *empirical loss* instead of the *expected loss* with respect to a specific loss (the $\log$ loss):

$$\theta^* = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, w_i)$$

Pre-historic languages

Imagine a language with two words: “argh” and “blah”

Pre-historic languages

What is Ω?
$\Omega = \{\text{argh}, \text{blah}\}$

What is Θ?
$\Theta = [0, 1]$

θ is the probability of “argh”

$1 - \theta$ is the probability of “blah”

What is the training data?
$w^{(1)} = \text{argh}$, $w^{(2)} = \text{argh}$, $w^{(3)} = \text{blah}$, $w^{(1)} = \text{argh}$, ...

Pre-historic languages

What is the likelihood objective function?
$p(w_i \mid \theta) = \theta$ if $w_i = \text{argh}$ and $1 - \theta$ if $w_i = \text{blah}$.

$p(w_i \mid \theta) = \theta^{I(w_i=\text{argh})}(1-\theta)^{I(w_i=\text{blah})}$
What is the log-likelihood objective?
$\log p(w_i \mid \theta) = I(w_i = \text{argh}) \log \theta + I(w_i = \text{blah}) \log(1-\theta)$
$L(w_1, \ldots, w_n \mid \theta) = \sum_{i=1}^n \log p(w_i \mid \theta) = \sum_{i=1}^n I(w_i = a) \log \theta + (1 - I(w_i = b) \log(1-\theta)$

$$= \underbrace{\left(\sum_{i=1}^n I(w_i = a)\right)}_{a} \log \theta + \underbrace{\left(\sum_{i=1}^n 1 - I(w_i = b)\right)}_{b} \log(1-\theta)$$

$$= a \log \theta + b \log(1-\theta)$$

Pre-historic languages

Log-likelihood: $L(\theta, w_1, \ldots, w_n) = a \log \theta + b \log(1 - \theta)$

The maximisation problem: $\theta^* = \arg\max_\theta L(\theta, w_1, \ldots, w_n)$
$\frac{\partial L}{\partial \theta} = \frac{a}{\theta} - \frac{1}{1-\theta} \times b$
Equate derivative to 0
$a(1 - \theta) - b\theta = 0$, note that $a + b = n$
Solution is
$\theta^* = \frac{a}{a+b} = \frac{a}{n}$

That's the maximum likelihood solution.

Principle of maximum likelihood estimation

- Objective function: log-likelihood (or likelihood)
- Estimation: maximise the log-likelihood with respect to the set of parameters

A guessing game

I choose a random number between 1 and 20. You need to guess it, and each time you make a guess I tell you whether your guess is higher or lower than my number. What is your strategy to guess the number as quickly as possible?

Binary search. Number of steps: $\log_2 n = -\log_2 \frac{1}{n}$.

A guessing game

I choose a random number between 1 and 20. You need to guess it, and each time you make a guess I tell you whether your guess is higher or lower than my number. What is your strategy to guess the number as quickly as possible?

Binary search. Number of steps: $\log_2 n = -\log_2 \frac{1}{n}$.

I choose a random number x between 1 and 20 **from a distribution** $p(x)$. You know p and need to guess the number. What is your strategy?

What does log-probability mean?

Let p be a probability distribution over Ω. What is $-\log_2 p(x)$?

Number of bits it takes to encode an optimal code for Ω when the true distribution is $p(x)$

Entropy:

$$H(p) = -\sum_x p(x) \log_2 p(x) = \mathbb{E}_p[|\text{code}(x)|]$$

The code is a bit-by-bit description of whether we take the decision "lower" or "higher" in the game

Another view of maximum likelihood estimation

What is the "empirical distribution?"
$\tilde{p}(w)$ be a probability distribution over the domain of datapoints such that $\tilde{p}(w)$ is the fraction of the n datapoints such that they are identical to w.

$$\tilde{p}(w) = \frac{\mathrm{count}(w; w^{(1)}, \ldots, w^{(n)})}{n}$$

Rewriting the objective function $L(\theta, w_1, \ldots, w_n)$

$$L(\theta, w_1, \ldots, w_n) = \frac{1}{n} \sum_{i=1}^{n} \log p(w_i \mid \theta)$$

$$= \sum_{w \in \Omega} \tilde{p}(w) \log p(w \mid \theta)$$

This is the cross entropy between $\tilde{p}$ and p

Cross-entropy

What is the definition of cross-entropy?

$$\mathrm{CE}(p,q) = -\sum_{x} p(x)\log q(x) = \mathbb{E}_p[-\log q(x)]$$

Cross-entropy

What is the definition of cross-entropy?

$$\mathrm{CE}(p,q) = -\sum_x p(x)\log q(x) = \mathbb{E}_p[-\log q(x)]$$

- Cross entropy is *not symmetric*, as such it is not “distance”, but it does tell whether p and q are close to each other
- For any given p, it is minimized when $q = p$
- It tells the expected number of bits we would use if we “encode” using q when p is the true distribution

Likelihood maximisation

By doing maximum likelihood maximisation we:

- Choose the parameters that make the data most probable,

  or, from an information-theoretic perspective:

- Choose the parameters that make the encoding of the data most succinct (bit-wise),

  in other words, we

- Minimize the cross-entropy between the empirical distribution and the model we choose.

Types of Models

It is often the case that we discuss a model $p(x \mid \theta)$

Really, in NLP, you are interested in predicting some $y(x)$

Therefore, you need $p(x, y \mid \theta)$. Estimation is the same when both x and y are in the dataset. Later we will learn about incomplete data

In some cases you model also $p(y \mid x, \theta)$ (e.g. neural networks, log-linear models).

This gives the generative vs. discriminative model distinction

Types of Objectives

We showed an example of deriving the log-likelihood solution for a simple model

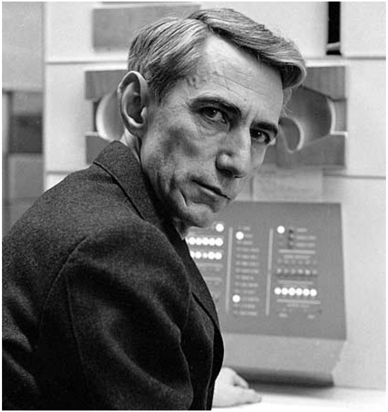
One can have more complex objective functions, and the principle would be the same

You just might not have a closed-form solution (e.g. with deep learning, log-linear models, etc.)

You need to apply an *optimisation* algorithm – more on that later

A bit of history

One of the earliest experiments with statistical analysis of language – measuring entropy of English



2-3 bits are required for English

Approach 2: the Bayesian approach

- History: 1700s. Seminal ideas due to Thomas Bayes and Pierre-Simon Laplace

- A lot has changed since then...