Natural Language Processing

Michael Noukhovitch

Fall 2020, McGill

Notes written from Jackie Cheung's lectures

Contents

1 Introduction

1.1 Overview

language is a form of communication

- *arbitrary* pairing between form and meaning
- very expressive and productive
- nearly universal
- uniquely human*

computational linguistics modelling natural language with computational models

- acoustic signals
- NL understanding (comprehension)
- NL generation (production)

goals of the field

- practical technologies (NLP)
- understanding how language works (CL)

models and techniques

- gathering data
- evaluation
- statistical methods (ML)
- rule-based systems

some example problems

- is language an instinct? (Chomsky)
- language processing to understand meaning of sentence
- can we learn mathematical properties of language

types of language

- **text** an idealization of spoken language
  - luckily English is similar between writing and speaking, and there is lots of data on it
  - older work used "clean" language but recent work ventures into messy data (e.g. Twitter)
- **speech** is much messier
  - automatic speech recognition (ASR)
  - text-to-speech generation (TTS)

1.2 Domains of Language

phonetics study of speech sounds

- articulation, transmission
- how each sound is made in the mouth

phonology rules that govern sound patterns

- how the sounds are organized
- "p" in peach and speach are the same phoneme but phonetically distinct (aspiration)

morphology word formation and meaning

- anti-dis-establish-ment-arian-ism

syntax structure of language

- "I a woman saw park in the" is **ungrammatical**
- **ambiguity** different possible meaning for the same phrase

semantics meaning of language

- "Ross wants to marry **a** Swedish woman"

pragmatics meaning of language in context

- different from literal meaning
- **deixis** interpretation that relies on extra-linguistic context
- "dessert would be delicious"

discourse structure of larger spans of language

- do large spans of text form a coherent story

1.3 Technology

combination of hand-crafted knowledge and ML on data

- rule-based systems
- machine learning
- knowledge representation

2 Text Classification

2.1 Basics

text classification assign a label or category to a piece of text

- sentiment analysis
- spam detection
- language identification
- authorship attribution

supervised output data is labelled

- learn a function, minimize θ with loss on data
- e.g. spam classificaiton, predict POS
- **regression** y is continuous
- **classification** y is discrete

unsupervised output data is unlabelled

- learn a density
- e.g. grammar induction, word-relatedness (word2vec)

2.2 Building a text classifier

- define problem, collect data
- extract feats
- train a classifier on train data
- apply classifier to test data

problem definition

- problem
- input
- output categories
- how to annotate

2.3 Feature Extraction

feature extraction get "important" properties of documents

- convert text into numerical format
- e.g. word counts as features *unigram counts*

lemma remove affixes get dictionary word "flies → fly" **stemming** remove affix get stem "airliner → airlin"

- rule-based e.g. (Porter, 1980) "ies → i"

n-grams sequences of adjacent words

- presence or absence
- counts
- proportion of total document
- scaled version (tf-idf)

POS tags crudely capture syntactic pattern (PTB dataset) **stop-word removal** remove common uninformative words

2.4 Models

training select parameters θ^* according to some objective
types of models

- **generative** models joint distribution $P(x, y)$
  - less flexible features as they need to be consistent with each other
- **discriminative** models conditional $P(y|x)$
  - can be more flexible in terms of features

2.4.1 Naive Bayes

Naive Bayes probabilistic classifier that uses Bayes' Rule $P(y|x) = \frac{P(y)P(x|y)}{P(x)}$

- generative
- assumes data x is generated independently conditioned on class $P(x_i|y)$
- graphical assumption $P(x, y) = P(y) \prod_i P(x_i|y)$

In NLP, we can assume NB over a *categorical* distribution and train

- loss $L = \prod_{(x,y)\in D} P(y) \prod_i P(x_i|y)$
- learn $P(Y = y)$ proportion of samples with class y
- learn $P(X_i = x|Y = y)$ proportion of samples with feature x given class y

Inference time we want $P(y|x)$

$$P(y|x) = P(x,y)|P(x) \tag{1}$$
$$= P(y)\prod_i P(x_i|y)/P(x) \tag{2}$$

where $P(x)$ is the marginalized over all classes
how to deal with multiple instances

- **type** identity of a word (count each word once)
- **token** instance of a word (count number of occurences)

2.4.2 Logistic Regression

logistic regression linear regression with a logit activation

- $P(y|x) = \frac{1}{Z}\exp(\sum_i a_i x_i)$
- squash output between $(0, 1)$

train log-likelihood with *gradient descent*

$$\log L(\theta) = \prod_{(x,y)\in D} \log P(y|x;\theta) \tag{3}$$
$$= \prod_{(x,y)\in D} (\sum_i a_i x_i - \log Z) \tag{4}$$

2.4.3 Support Vector Machines

SVM learns linear decision to maximize margin to nearest sample in each of two classes

- can be non-linear using *kernels*

2.4.4 Neural Network

Perceptron logistic regression with Perceptron learning rule $f(x) = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{else} \end{cases}$

Stacked Perceptron stacks perceptron neurons
Artificial Neural Network stacked neurons with non-linear activation functions

- can learn complex functions
- need lots of data and computational power

2.5 Model Selection

How to choose preprocessing, model, etc.. evaluate on unseen data!
Data split

- **training** learning the model, 60-90%
- **dev/validation** evaluating while learning the model
- **testing** evaluate once at the end to see how well you do

k-fold cross-validation split training data into k folds, train on $k-1$ fold and test on the last
key issues

- which eval measure to use
- stastical significance of test
- do these tests matter?