Feature Extraction

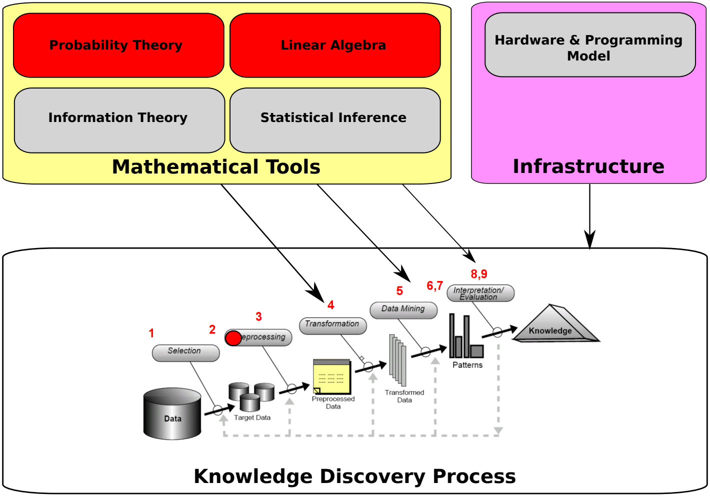
Knowledge Discovery and Data Mining 1

Roman Kern

KTI, TU Graz

2014-10-23

Big picture: KDDM



Outline

1. Introduction
2. Feature Extraction from Text

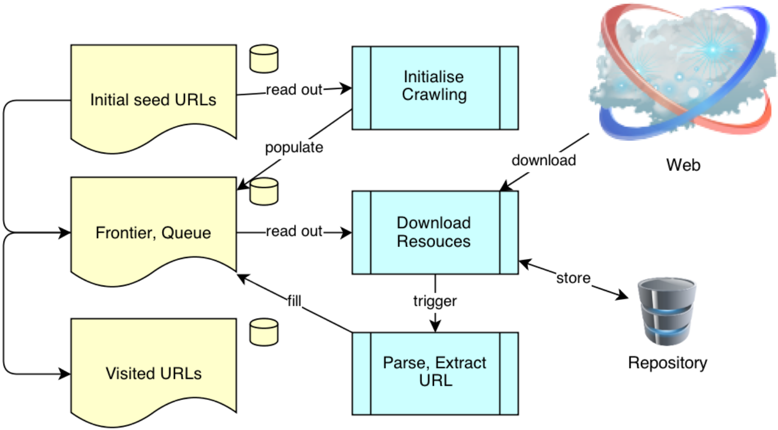
Recap

Review of the preprocessing phase

Introduction

- Initial phase of the Knowledge Discovery process
- ... acquire the data to be analysed
- e.g. by **crawling** the data from the Web
- ... prepare the data
- e.g. by **cleaning** and **removing outliers**

Simple Web crawling schema



Web information extraction

- Web information extraction is the problem of extracting target information item from Web pages
- $\rightarrow$ Two problems
  1. Extract information from natural language text
  2. Extract structured data from Web pages

- Three basic approaches for wrapper generation:
  1. Manual - simple approach, but does not scale for many sites
  2. Wrapper induction - supervised approach
  3. Automatic extraction - unsupervised approach

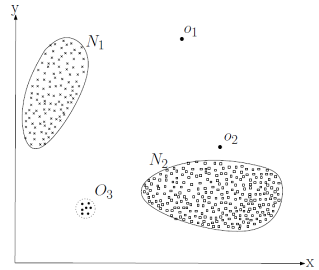
Data cleaning

- Often data sets will contain:
  - Unnecessary data
  - Missing values
  - Noise
  - Incorrect data
  - Inconsistent data
  - Formatting issues
  - Duplicate information
  - Disguised data
- These factors will have an impact on the results of the data mining process

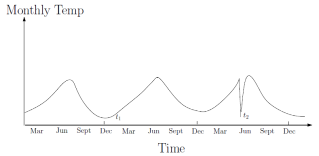
Garbage in $\rightarrow$ garbage out

Types of outliers

Point outliers



Contextual outliers



Collective outliers

Feature Extraction

What are features?

Introduction

Data vs. Information

- Raw data is useless
- Need techniques to (automatically) extract information from it
- Data: recorded (collected, crawled) facts
- Information: (novel, informative, implicit, useful, ...) patterns within the data

Introduction

What are features?

- An individual measurable property of a phenomenon being observed
- The items, that represent knowledge suitable for Data Mining algorithms
- A piece of information that is potentially useful for prediction

They are sometimes also called *attributes* (Machine Learning) or *variables* (statistics).

Introduction

Example of features:

- Images → colours, textures, contours, ...
- Signals → frequency, phase, samples, spectrum, ...
- Time series → ticks, trends, self-similarities, ...
- Biomed → dna sequence, genes, ...
- Text → words, POS tags, grammatical dependencies, ...

Features encode these properties in a way suitable for a chosen algorithm

Introduction

Types of Features

- Numeric (for quantitative data)
  - Continuous, e.g. height, time, ...
  - Discrete, e.g. counts
- Categorial (for qualitative data, level of measurement [Stevens 1946])
  - Nominal
    - Two or more categories
    - e.g. gender, colour
  - Ordinal
    - There is an ordering within the values
    - e.g. ranking
    - Interval, if intervals are equally split, e.g. Likert scale, date
    - Ratio, for intervals with a defined zero point, e.g. temperature, age

Binary features are quite common - what are they?

Introduction

Categories of Features

- Contextual features
  - e.g. n-grams, position information
- Stuctural features
  - e.g. structural markups, DOM elements
- Linguistic features
  - e.g. POS tags, noun phrases
- ...

Introduction

Example for feature extraction

- Handwriting recognition
- ... popular introductory example in textbooks about machine learning, e.g. Machine Learning in Action [Harrington 2012]

Introduction

Example for feature extraction

- Input: A collection of scanned in handwritten digits
- Preprocessing:
    - Remove noise
    - Adapt saturation changes, due to differences in pressure when writing
    - Normalise to the same size
    - Center the images, e.g. center of mass or bounding box
- Feature extraction:
    - Pixels as binary features

Depending on the algorithm to center the images, some algorithm improve in performance, e.g. SVM according to the authors of the MNIST data set

Text mining

Introduction

Text mining

=

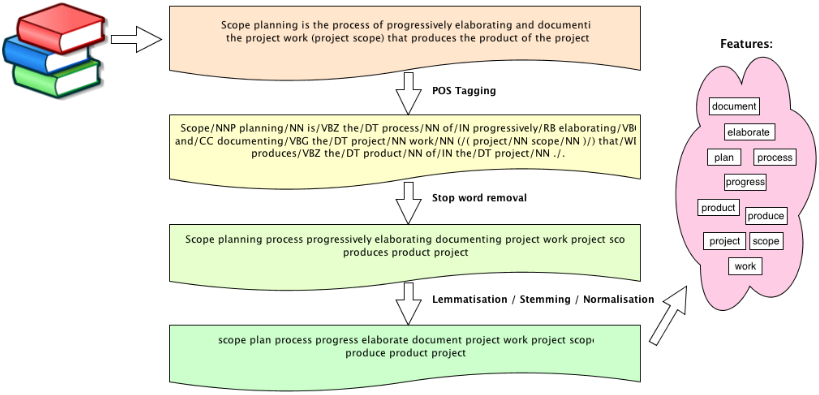
data mining (applied to text data)

+

basic linguistics

Text Mining is understood as a process of automatically extracting meaningful, useful, previously unknown and ultimately comprehensible information from textual document repositories.

Text mining - example pipeline



Feature Extraction from Text

Example: Part-of-Speech Tagging

POS - Introduction

What is Part-of-Speech?

- The process to apply word classes to words within a sentence
- For example
  - Car → *noun*
  - Writing → *noun* or *verb*
  - Grow → *verb*
  - From → *preposition*

Open vs closed word classes

- Prepositions (closed, e.g. “of”, “to”, “in”)
- Verbs (open, e.g. to “google”)

POS - open classes

Open classes

Four main open classes:

- Nouns
- Verbs
- Adjectives
- Adverbs

POS - open classes

Nouns

- Proper nouns
  - e.g. names of persons or entities, e.g. Linux
- Common nouns
  - Count nouns, can be enumerated, e.g. one goat
  - Mass nouns, conceptualised as a homogeneous group, e.g. snow

Adjectives

- Adjectives for concepts such as
  - Color, age, value and others

POS - open classes

Verbs

- non-3rd-person-singular (eat)
- 3rd-person-singular (eats)
- Progressive (eating)
- Past participle (eaten)

Adverbs

- Modifying "something" (often verbs)
- *Unfortunately*, John walked home extremely slowly yesterday
- Directional, locative, degree, manner and temporal adverbs

POS - closed classes

Closed classes

Main classes:

- Prepositions
- Determiners
- Pronouns
- Conjunctions
- Auxiliary verbs
- Particles
- Numerals

POS - closed classes

Preposition

- Occur before noun phrases, often indicating spatial or temporal relations
- `on`, `under`, `over`, `near`, `by`, `at`, `from`, `to`, `with`

Determiner ("Artikelwörter")

- `a`, `an`, `the`

POS - closed classes

Pronoun

- Often act as a kind of shorthand for referring to some noun phrase or entity or event
- `she, who, I, others`

Conjunctions ("Bindewörter")

- Used to join two phrases, clauses or sentences
- `and, but, or, as, if, when`

POS - closed classes

Auxiliary verbs ("Hilfsverben")

- Mark whether an action takes place in the present, past or future, whether it is completed, whether it is negated and whether an action is necessary, possible, suggested or desired
- `can, may, should, are`

Particles ("Verbindungswörter")

- A word that resembles a preposition or an adverb, often combines with a verb to form a larger unit (went on, throw off, etc.)
- `up, down, on, off, in, out, at, by, into, onto`

Numerals

- `one, two, three, first, second, third`

POS tagging

What is POS tagging?

Part-of-speech tagging is the process of assigning a part-of-speech or other lexical class marker to each word in a corpus [Jurafsky & Martin]

POS tagging process

- Input: a string of words and a specified tagset
- Output: a single best match for each word

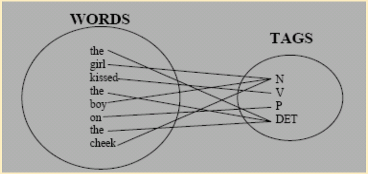


Figure: Assing words to tags out of a tagset [Jon Atle Gulla]

POS tagging

Examples:

- `Book that flight.`
- `VB DT NN`

- `Does that flight serve dinner?`
- `VBZ DT NN VB NN`

This task is not trivial

- For example: "book" is ambiguous (noun or verb)
- Challenge for POS tagging: resolve these ambiguities!

POS tagging - tagsets

Tagset

The tagset is the vocabulary of possible POS tags

Choosing a tagset

Striking a balance between

- Expressiveness (number of different word classes)
- "Classifiability" (ability to automatically classify words into the classes)

POS tagging - tagsets

Examples for existing tagsets:

- Brown corpus, 87-tag tagset (1979)
- Penn Treebank, 45-tag tagset, selected from Brown tagset (1993)
- C5, 61-tag tagset
- C7, 146-tag tagset
- STTS, German tagset (1995/1999)
  http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html

POS tagging

The Brown corpus

- 1 mio words of American English texts, printed in 1961
- Sampled from 15 different text categories
- The first, and for a long time the only, modern, computer readable general corpus.
- The Corpus is divided into 500 samples of 2000+ words each.
- The samples represent a wide range of styles and varieties of prose.
  - General fiction, mystery, science fiction, romance, humour,
  - Sources books, newspapers, magazines, ...
- Does not include the tagset, the “Brown Corpus Tagset” represents a tagset that has been applied to the Brown Corpus
- http://icame.uib.no/brown/bcm.html

POS tagging

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Figure: Penn Treebank POS tags

POS tagging

Penn Treebank

- Over 4.5 mio words
- Presumed to be the first large syntactically annotated corpus
- Annotated with POS information
- And with skeletal syntactic structure

Two-stage tagging process:

1. Assigning POS tags automatically (stochastic approach, 3-5% error)
2. Correcting tags by human annotators

POS tagging

Table 4:
Penn Treebank
(as of 11/92)

Description	**Tagged for Part-of-Speech** (Tokens)	**Skeletal Parsing** (Tokens)
Dept. of Energy abstracts	231,404	231,404
Dow Jones Newswire stories	3,065,776	1,061,166
Dept. of Agriculture bulletins	78,555	78,555
Library of America texts	105,652	105,652
MUC-3 messages	111,828	111,828
IBM Manual sentences	89,121	89,121
WBUR radio transcripts	11,589	11,589
ATIS sentences	19,832	19,832
Brown Corpus, retagged	1,172,041	1,172,041
Total:	4,885,798	2,881,188

Figure: Penn Treebank POS corpus

POS tagging

How hard is the tagging problem?

Unambiguous (1 tag)	**35,340**	
Ambiguous (2–7 tags)	**4,100**	
2 tags	3,760	
3 tags	264	
4 tags	61	
5 tags	12	
6 tags	2	
7 tags	1	("still")

Figure: The number of word classes in the the Brown corpus by degree of ambiguity

POS tagging

Main approaches for POS tagging

- Rule based
  - ENGTWOL tagger
- Transformation based
  - Brill tagger
- Stochastic
  - HMM tagger

POS tagging

Rule based POS tagging

- A two stage process
  1. Assign a list of potential parts-of-speech to each word, e.g. BRIDGE $\rightarrow$ V N
  2. Using rules, eliminate parts-of-speech tags from that list until a single tag remains
- ENGTWOL uses about 1.100 rules to rule out incorrect parts-of-speech

POS tagging

Input

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**
...	

Rules

ADVERBIAL-THAT RULE
Given input: "that"
if
(+1 A/ADV/QUANT); /* *if next word is adj, adverb, or quantifier* */
(+2 SENT-LIM); /* *and following which is a sentence boundary,* */
(NOT -1 SVOC/A); /* *and the previous word is not a verb like* */
/* *'consider' which allows adjs as object complements* */
then eliminate non-ADV tags
else eliminate ADV tag

POS tagging

Transformation based POS tagging

- Brill Tagger [Brill 1995]
- Combination of rule-based tagger with supervised learning
- Rules:
    - Initially assign each word a tag (without taking the context into account)
        - Known words → assign the most frequent tag
        - Unknown word → e.g. noun (guesser rules)
    - Apply rules iteratively (taking the surrounding context into account → context rules)
    - e.g. `If Trigger, then change the tag from X to Y`,
    - `If Trigger, then change the tag to Y`
- Typically 50 guessing rules and 300 context rules
- Rules have been induced from tagged corpora by means of Transformation-Based Learning (TBL)

http://www.ling.gu.se/~/lager/mogul/brill-tagger/index.html

POS tagging

Transformation-Based Learning - based on tagged training data set

1. Generate all rules that correct at least one error
2. For each rule:
    1. Apply a copy of the most recent state of the training set
    2. Score the result using the objective function (e.g. number of wrong tags)
3. Select the rules with the best score
4. Update the training set by applying the selected rules
5. Stop if the the score is smaller than some pre-set threshold T; otherwise repeat from step 1

POS tagging

Stochastic part-of-speech tagging

- Based on probability of a certain tag given a certain context
- Necessitates a training corpus
- No probabilities available for words not in training corpus
  - Smoothing
- **Simple Method**: Choose the most frequent tag in the training text for each word
  - Result: 90% accuracy
  - Baseline method
- Lot of non-trivial methods, e.g. **Hidden Markov Models (HMM)**

POS tagging - Stochastic part-of-speech tagging

Motivation

- Statistical NLP aims to do statistical inference for the field of NL
- *Statistical inference* consists of taking some data (generated in accordance with some unknown probability distribution) and then making some inference about this distribution.
- An example of statistical inference is the task of language modelling (ex how to predict the next word given the previous words)
- In order to do this, we need a model of the language.
- Probability theory helps us finding such model

POS tagging - Stochastic part-of-speech tagging

The noisy channel model

- Given an input stream of data, which gets corrupted in a **noisy channel**
- Assume, the input has been a string of words with their associated POS tags
- The output we observe is a string of words
- Word+POS $\rightarrow$ noisy channel $\rightarrow$ word
- The task is to recover the missing POS tag

POS tagging - Stochastic part-of-speech tagging

Markov models & Markov chains

- Markov chains can be seen as a weighted finite-state machines
- They have the following Markov properties, where X_i is a state in the Markov chain, and s is a value that the state takes:
  - **Limited horizon**: $P(X_{t+1} = s|X_1, ..., X_t) = P(X_{t+1} = s|X_t)$ (first order Markov models)
  - ... the value at state $t + 1$ just depends on the previous state
  - **Time invariant**: $P(X_{t+1} = s|X_t)$ is always the same, regardless of t
  - ... there are no side effects

POS tagging - Stochastic part-of-speech tagging

Example of a **transition matrix** (A) corresponding to a Markov model for word sequences involving: *the*, *dogs*, *bit*:

	the	dogs	bit
the	0.01	0.46	0.53
dogs	0.05	0.15	0.80
bit	0.77	0.32	0.01

$P(dogs|the) = 0.46$ ... the probability of word *dogs* to follow *the* is 46%.

Example of a **initial probability matrix** (π):

the	0.7
dogs	0.2
bit	0.1

Note: The A matrix can be seen as bi-gram Language Model and π as unigram Language Model.

POS tagging - Stochastic part-of-speech tagging

- What is the probability of the sequence "*the dogs bit*"?
- → multiply the probabilities:
  - $P(the, dogs, bit) = \pi(the) * A(dogs|the) * A(bit|dogs) = 0.7 * 0.46 * 0.80 = 0.2576$
- What is the probability of *dogs* as the second word?
- → add the probabilities:
  - $p(w_2 = dogs) = \pi(the) * A(dogs|the) + \pi(dogs) * A(dogs|dogs) + \pi(bit) * A(dogs|bit)$

If we have the probability of the other two words (the, bit) as second word, we can determine which is the best second word.

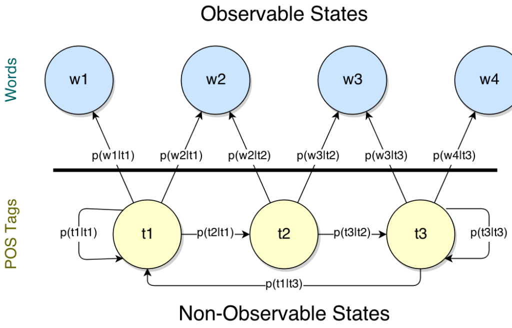
POS tagging - Stochastic part-of-speech tagging

Hidden Markov Models

- Now, that we are given a sequence of words (observation) and want to find the POS tags?
  - Each state in the Markov model will be a POS tag (hidden state), but we don't know the correct state sequence
  - The underlying sequence of events (= the POS tags) can be seen as generating a sequence of words
  - ... thus, we have a Hidden Markov Model
- $\Rightarrow$ Requires an additional emission matrix (B), linking words to POS tags

POS tagging - Stochastic part-of-speech tagging

Hidden Markov Models



Needs three matrices as input: A (transmission, POS $\mapsto$ POS), B (emission, POS $\mapsto$ Word), π (initial probabilities, POS)

POS tagging - Stochastic part-of-speech tagging

Hidden states: DET, N, and VB
... then the **transmission matrix** (A - POS $\rightarrow$ POS) could look like:

	DET	N	VB
DET	0.01	0.89	0.10
N	0.30	0.20	0.50
VB	0.67	0.23	0.10

... **emission matrix** (B - POS $\rightarrow$ word):

	the	dogs	bit	chased	a	these	cats	...
DET	0.33	0.0	0.0	0.0	0.33	0.33	0.0	...
N	0.0	0.2	0.1	0.0	0.0	0.0	0.15	...
VB	0.0	0.1	0.6	0.3	0.0	0.0	0.0	...

... **initial probability matrix** (π):

DET	0.7
N	0.2
VB	0.1

POS tagging - Stochastic part-of-speech tagging

Generative model

- In order to generate sequence of words, we:
  1. Choose tag/state from π
  2. Choose emitted word from corresponding row of B
  3. Choose transition from corresponding row of A
  4. GOTO 2 (while keeping track of the probabilities)
- This is easy, as the state stays known
- If we wanted, we could generate all possibilities this way and find the most probable sequence

POS tagging - Stochastic part-of-speech tagging

State sequences

- Given a sequence of words, we don't know with tag sequence generated it, e.g. "the bit dogs"
  - `DET N VB`
  - `DET N N`
  - `DET VB N`
  - `DET VB VB`
- Each tag sequence has different probabilities
- $\rightarrow$ we need an algorithm which will give us the best sequence of states (i.e. tags) for a given sequence of words

POS tagging - Stochastic part-of-speech tagging

Three fundamental problems

1. **Probability estimation**: How do we efficiently compute probabilities, i.e. $P(O|\mu)$ - the probability of an observation sequence O given a model μ
   - $\mu = (A, B, \pi)$, A ... transition matrix, B ... emission matrix, π initial probability matrix
2. **Best path estimation**: How do we choose the best sequence of states X, given our observation O and the model μ
   - How do we maximise $P(X|O)$?
3. **Parameter estimation**: From a space of models, how do we find the best parameters (A, B, and π) to explain the observation
   - How do we (re)estimate μ in order to maximise $P(O|\mu)$?

POS tagging - Stochastic part-of-speech tagging

Three fundamental problems

1. **Probability estimation**
   - Dynamic programming (summing forward probabilities)
2. **Best path estimation**
   - Viterbi algorithm
3. **Parameter estimation**
   - Baum-Welch algorithm (Forward-Backward algorithm)

POS tagging - Stochastic part-of-speech tagging

Simplifying the probabilities

- $\text{argmax}_{t_{1,n}} P(t_{1,n}|w_{1,n}) = \text{argmax}_{t_{1,n}} P(w_{1,n}|t_{1,n})P(t_{1,n})$
- $\rightarrow$ refers to the whole sentence
- ... estimating probabilities for an entire sentence is a bad idea
- Markov models have the property of limited horizon: one state refers only back the previous (n, typically 1) steps - is has no memory
- ... other assumptions

POS tagging - Stochastic part-of-speech tagging

Simplifying the probabilities

- Independence assumption: words/tags are independent of each other
  - For a bi-gram model:
  - $P(t_{1,n}) \approx P(t_n|t_{n-1})P(t_{n-1}|t_{n-2})...P(t_2|t_1) = \prod_{i=1}^{n} P(t_i|t_{i-1})$
- A word's identity only depends on its tag
  - $P(w_{1,n}|t_{1,n}) \approx \prod_{i=1}^{n} P(w_i|t_i)$
- The final equation is:
- $\hat{t}_{1,n} = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$

POS tagging - Stochastic part-of-speech tagging

Probability estimation for tagging

- How do we get such probabilities?
- $\rightarrow$ With supervised tagging we can simply use **Maximum Likelihood Estimation (MLE)** and use counts (C) from a reference corpus
  - $P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$
  - $P(w_i|t_i) = \frac{C(w_i, t_i)}{C(t_i)}$
- Given these probabilities we can finally assign a probability to a sequence of states (tags)
- To find the best sequence (of tags) we can apply the Viterbi algorithm

There is a IPython notebook for playing around with HMMs

POS tagging - Stochastic part-of-speech tagging

Probability estimation

- Given an observation, estimate the underlying probability
- e.g. recall PMF for binomial: $p(k) = \binom{n}{k}(1-p)^{n-k}p^k$
- We want to estimate the best p:
- $\text{argmax}_p P(\textit{observed data}) = \text{argmax}_p \binom{n}{k}(1-p)^{n-k}p^k$
- $\rightarrow$ derivative to find the maxima ($0 = \frac{\partial}{\partial p}\binom{n}{k}(1-p)^{n-k}p^k$)
- For large np one can approximate p to be $\frac{k}{n}$ (and standard deviation of $\sqrt{\frac{k(n-k)}{n^3}}$ for independent and an unbiased estimate)

There are alternative versions on how to estimate the probabilities

POS tagging - Stochastic part-of-speech tagging

- Does work for cases, where there is evidence in the corpus
- But what to do, if there are rare events, which just did not make it into the corpus?
- Simple non-solution: always assume their probability to be 0
- Alternative solution: **smoothing**

POS tagging - Stochastic part-of-speech tagging

Will the sun rise tomorrow?

- Laplace's Rule of Succession
- We start with the assumption that rise/non-rise are equally probable
- On day $n+1$, we've observed that the sun has risen s times before
- $p_{Lap}(S_{n+1} = 1 | S_1 + ... + S_n = s) = \frac{s+1}{n+2}$
- What is the probability on day 0, 1, ...?

POS tagging - Stochastic part-of-speech tagging

Laplace Smoothing

- Simply add one:
- $\frac{C(t_{i-1},t_i)}{C(t_{i-1})} \Rightarrow \frac{C(t_{i-1},t_i)+1}{C(t_{i-1})+V(t_{i-1},t)}$
- ... where $V(t_{i-1}, t) = |\{t_i | C(t_{i-1}, t_i) > 0\}|$ (vocabulary size)
- Can be further generalised by introducing a smoothing parameter λ
- $\frac{C(t_{i-1},t_i)+\lambda}{C(t_{i-1})+\lambda V(t_{i-1},t)}$

Also called Lidstone smoothing, additive smoothing

POS tagging - Stochastic part-of-speech tagging

Estimate the smoothing parameter

- $\frac{C(t_{i-1},t_i)+\lambda}{C(t_{i-1})+\lambda V(t_{i-1},t)}$
- ... typically λ is set between 0 and 1
- How to choose the correct λ?
- Separate a small part of the training set (held out data)
    - ... development set
- Apply the maximum likelihood estimate

POS tagging - Stochastic part-of-speech tagging

State-of-the-Art

System name	Short description	All tokens	Unknown words
TnT	Hidden markov model	96.46%	85.86%
MElt	MEMM	96.96%	91.29%
GENiA Tagger	Maximum entropy	97.05%	Not available
Averaged Perceptron	Averaged Perception	97.11%	Not available
Maxent easiest-first	Maximum entropy	97.15%	Not available
SVMTool	SVM-based	97.16%	89.01%
LAPOS	Perceptron based	97.22%	Not available
Morče/COMPOST	Averaged Perceptron	97.23%	Not available
Stanford Tagger 2.0	Maximum entropy	97.32%	90.79%
LTAG-spinal	Bidirectional perceptron	97.33%	Not available
SCCN	Condensed nearest neighbor	97.50%	Not available

Taken from:

http://aclweb.org/aclwiki/index.php?title=POS_Tagging_%28State_of_the_art%29

Thank You!

Next up: Feature Engineering