Word Disambiguation

Lecture #8

Introduction to Natural Language Processing
CMPSCI 585, Fall 2004

University of Massachusetts Amherst

Andrew McCallum

Words and their meaning

Three lectures:

- *Last time:* Collocations
  - multiple words together, different meaning than than the sum of its parts
- *Today:* Word disambiguation
  - one word, multiple meanings
  - Expectation Propagation
- *Future:* Word clustering
  - multiple words, "same" meaning

Today's Main Points

- What is word sense disambiguation, and why is it useful.
  - Homonymy, Polysemy
  - Other similar NLP problems
- 4 Methods for performing WSD.
  - Supervised, naïve Bayes
  - Unsupervised, Expectation Propagation

Word Sense Disambiguation

- The task is to determine which of various senses of a word are invoked in context.
  - *True annuals are **plants** grown from seed that blossom, set new seed and die in a single year.*
  - *Nissan's Tennessee manufacturing **plant** beat back a United Auto Workers organizing effort with aggressive tactics.*
- This is an important problem: Most words are ambiguous (have multiple senses)
- Problem statement:
  - A word is assumed to have a finite number of discrete senses
  - Make a forced choice between each word usage based on some limited context around the word
- Converse: word or senses that mean (almost) the same:
  - image, likeness, portrait, facsimile, picture
  - (Next lecture)

WSD important for…

- Translation
  - "The spirit is willing but the flesh is weak."
  - "The vodka is good, but the meat is spoiled."
- Information Retrieval
  - query: "wireless mouse"
  - document: "Australian long tail hopping mouse"
- Computational lexicography
  - To automatically identify multiple definitions to be listed in a dictionary
- Parsing
  - To give preference to parses with correct use of senses
- There isn't generally one way to divide the uses of a word into a set of non-overlapping categories.
- Senses depend on the task [Kilgarriff 1997]

WSD: Many other cases are harder

- "*title*"
  - Name/heading of a book, statute, work of art of music, etc.
  - Material at the start of a film
  - The right of legal ownership (of land)
  - The document that is evidence of this right
  - An appellation of respect attached to a person's name
  - A written work

WSD: types of problems

- Homonymy: meanings are unrelated:
  - ***bank*** *of a river*
  - ***bank*** *financial institution*
- Polysemy: related meanings (as on previous slide)
  - **title** of a book
  - **title** material at the start of a film
- Systematic polysemy: standard methods of extending a meaning, such as from an organization to the building where it is housed.
  - *The speaker of the* ***legislature****…*
  - *The* ***legislature*** *decided today…*
  - *He opened the door, and entered the* ***legislature***
- A word frequently takes on further related meanings through systematic polysemy or metaphor.

Upper and lower bounds on performance

- Upper bound: human performance
  - How often do human judges agree on the correct sense assignment?
  - Particularly interesting if you only give humans the same input context given to machine method.
    (A good test for any NLP method!)
  - Gale 1992: give pairs of words in context, humans say if they are the same sense. Agreement 97-99% for word with clear senses, but ~65-70% for polysemous words.
- Lower bound: simple baseline algorithm
  - Always pick the most common sense for each word.
  - Accuracy depends greatly on sense distribution! 90-50%?

Senseval competitions

- Senseval 1: September 1998. Results in *Computers and the Humanities* 34(1-2). OUP Hector corpus.
- Senseval 2: In first half of 2001. WordNet senses.
  - http://www.itri.brighton.ac.uk/events/senseval

WSD automated method performance

- Varies widely depending on how difficult the disambiguation task is.
- Accuracies over 90% are commonly reported on the classic, often fairly easy, word disambiguation tasks (pike, star, interest)
- Senseval brought careful evaluation of difficult WSD (many senses, different POS)
- Senseval 1, fine grained senses, wide range of types
  - Overall: about 75% accuracy
  - Nouns: about 80% accuracy
  - Verbs: about 70% accuracy

WSD solution #1: expert systems
[Small 1980] [Hirst 1988]

- Most early work used semantic networks, frames, logical reasoning, or "expert system" methods for disambiguation based on contexts.
- The problem got quite out of hand:
  - The word expert for "throw" is "currently six pages long, but should be ten times that size" (Small and Rieger 1982)

WSD solution #2: dictionary-based
[Lesk 1986]

- A word's dictionary definitions are likely to be good indicators for the senses they define.
  - One sense for each dictionary definition
  - Look for overlap between words in definition and words in context at hand

Word="ash"

Sense	Definition
1. tree	a tree of the olive family
2. burned	the solid residue left when combustible material is burned

"This cigar burns slowly and creates a stiff ***ash****"* sense1=0 sense2=1
"The ***ash*** *is one of the last trees to come into leaf"* sense1=1 sense2=0

- Insufficient information in definitions. Accuracy 50-70%

WSD solution #3: thesaurus-based
[Walker 1987] [Yarowsky 1992]

- Occurrences of a word in multiple thesaurus "subject codes" is a good indicator of its senses.
- Count number of times context words appear among the entries for each possible "subject code".
- Increase coverage of rare words and proper nouns by also looking in the thesaurus for words that co-occur with context words more often than chance. E.g. "Hawking" co-occurs with "cosmology", "black hole"

Word	Sense	Roget category	Accuracy
star	space object	UNIVERSE	96%
	celebrity	ENTERTAINER	95%
	star-shaped	INSIGNIA	82%

An extra trick: global constraints
[Yarowsky 1995]

- One sense per discourse: the sense of a word is highly consistent within a document
  - Get a lot more context words because combine the context of multiple occurrences
  - True for topic dependent words
  - Not so true for other items like adjectives and verbs, e.g. "*make*", "*take*".

Other similar "disambiguation" problems

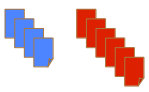
- Sentence boundary detection
  - *"I live on Palm Dr. Smith lives downtown."*
  - Only really ambiguous when
    - word before the period is an abbreviation (which can end a sentence - not something like a title)
    - word after the period is capitalized (and can be a proper name - otherwise it must be a sentence end)
- Context-sensitive spelling correction
  - *"I know their is a problem with there account."*

WSD solution #4: supervised classification

- Gather a lot of labeled data: words in context, hand-labeled into different sense categories.
- Use naïve Bayes document classification with "context" as the document!
  - Straightforward classification problem.
  - Simple, powerful method! :-)
  - Requires hand-labeling a lot of data :-(
- Can we still use naïve Bayes, but without labeled data?…

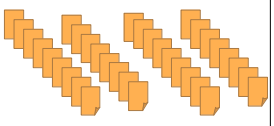
WSD sol'n #5: unsupervised disambiguation

word+context, labeled according to sense



Train one multinomial per class via maximum likelihood.
What you just did for HW#1

word+context, unlabeled



Label is missing!

28 years ago…

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: Maximum likelihood; incomplete data; EM algorithm; posterior mode

1. Introduction

This paper presents a general approach to iterative computation of maximum-likelihood estimates when the observations can be viewed as incomplete data. Since each iteration of the algorithm consists of an expectation step followed by a maximization step we call it the EM algorithm. The EM process is remarkable in part because of the simplicity and generality of

Filling in Missing Labels with EM

[Dempster *et al* '77], [Ghahramani & Jordan '95], [McLachlan & Krishnan '97]

Expectation Maximization is a class of iterative algorithms for maximum likelihood estimation with incomplete data.

- ***E-step:*** Use current estimates of model parameters to "guess" value of missing labels.
- ***M-step:*** Use current "guesses" for missing labels to calculate new estimates of model parameters.
- Repeat E- and M-steps until convergence.

Finds the model parameters that locally maximize the probability of both the labeled and the unlabeled data.

Recall: "Naïve Bayes"

Pick the most probable class, given the evidence:

$$c^* = \arg\max_{c_j} \Pr(c_j \mid d)$$

c_j - a class (like "Planning")

d - a document (like "language intelligence proof...")

Bayes Rule:

$$\Pr(c_j \mid d) = \frac{\Pr(c_j)\Pr(d \mid c_j)}{\Pr(d)}$$

"Naïve Bayes":

$$\approx \frac{\Pr(c_j)\prod_{i=1}^{|d|}\Pr(w_{d_i} \mid c_j)}{\sum_{c_k}\Pr(c_k)\prod_{i=1}^{|d|}\Pr(w_{d_i} \mid c_k)}$$

w_{d_i} - the i th word in d (like "proof")

Recall: Parameter Estimation in Naïve Bayes

Estimate of P(c)

$$P(c_j) = \frac{1 + \text{Count}(d \in c_j)}{|C| + \sum_k \text{Count}(d \in c_k)}$$

Estimate of P(w|c)

$$\hat{P}(w_i \mid c_j) = \frac{1 + \sum_{d_k \in c_j} \text{Count}(w_i, d_k)}{|V| + \sum_{t=1}^{|V|}\sum_{d_k \in c_j} \text{Count}(w_t, d_k)}$$

EM Recipe

- Initialization
  - Create an array P(c|d) for each document, and fill it with random (normalized) values. Set P(c) to the uniform distribution.
- M-step (likelihood **M**aximization)
  - Calculate maximum-likelihood estimates for parameters P(w|c) ***using current P(c|d).***

$$\hat{P}(w_i \mid c_j) = \frac{1 + \sum_{d_k \in D} \text{Count}(w_i, d_k) \cdot P(c_j \mid d_k)}{|V| + \sum_{t=1}^{|V|}\sum_{d_k \in D} \text{Count}(w_t, d_k) \cdot P(c_j \mid d_k)}$$

- E-step (missing-value **E**stimation)
  - ***Using current parameters***, calculate new P(c|d) the same way you would at test time.

$$\Pr(c_j \mid d) = \frac{\Pr(c_j)\Pr(d \mid c_j)}{\Pr(d)}$$

- Loop back to M-step, until convergence.
  - Converged when maximum change in a parameter P(w|c) is below some threshold.

EM

- We could have simply written down likelihood, taken derivative and solved…
  - but unlike "complete data" case, not solvable in closed form
  - must use iterative method:
  - gradient ascent
  - EM is another form of ascent on this likelihood surface
  - Convergence, speed and local minima are all issues.
- If you make "hard 0 versus 1" assignments in P(c|d), you get the K-means algorithm.
- Likelihood will always be highest with more classes.
  - Use a prior over number of classes, or just pick arbitrarily.

EM

- Some good things about EM
  - no learning rate parameter
  - very fast for low dimensions
  - each iteration is guaranteed to improve likelihood
  - adapts unused units rapidly
- Some bad things about EM
  - can get stuck in local minima
  - ignores model cost (how many classes?)
  - both steps require considering *all* explanations of the data (all classes)

Semi-Supervised Document Classification

Training data with class labels

Data available at training time, but without class labels

Web pages user says are interesting

Web pages user says are uninteresting

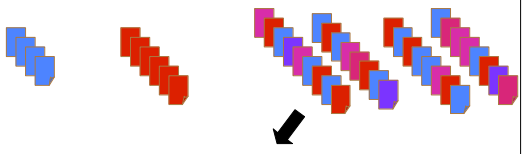
Web pages user hasn't seen or said anything about

Can we use the unlabeled documents to increase accuracy?

Semi-Supervised Document Classification

Build a classification model using limited labeled data

Use model to estimate the labels of the unlabeled documents



Use *all* documents to build a new classification model, which is often more accurate because it is trained using more data.

An Example

Labeled Data

Unlabeled Data

Baseball

The new hitter struck out...

Struck out in last inning...

Homerun in the first inning...

Pete Rose is not as good an athlete as Tara Lipinski...

Before EM:

Pr (Lipinski **)** = 0.01

Ice Skating

Fell on the ice...

Perfect triple jump...

Katarina Witt's gold medal performance...

New ice skates...

Practice at the ice rink every day...

Pr (Lipinski **)** = 0.001

Tara Lipinski's substitute ice skates didn't hurt her performance. She graced the ice with a series of perfect jumps and won the gold medal.

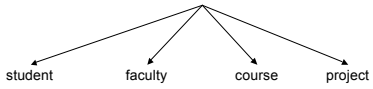
Tara Lipinski bought a new house for her parents.

After EM:

Pr (Lipinski | Ice Skating **)** = 0.02

Pr (Lipinski | *Baseball* **)** = 0.003

WebKB Data Set



4 classes, 4199 documents

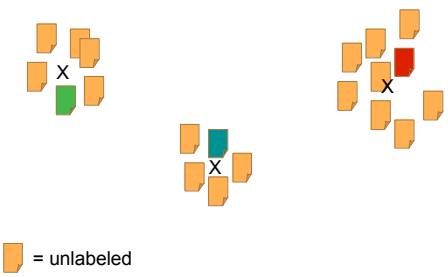
from CS academic departments

Word Vector Evolution with EM

Iteration 0	**Iteration 1**	**Iteration 2**
intelligence	*DD*	*D*
DD	*D*	*DD*
artificial	lecture	lecture
understanding	cc	cc
*DD*w	*D**	*DD:DD*
dist	*DD:DD*	due
identical	handout	*D**
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	*DD*am	hw
cognitive	yurtas	exam
logic	homework	problem
proving	kfoury	*DD*am
prolog	sec	postscript

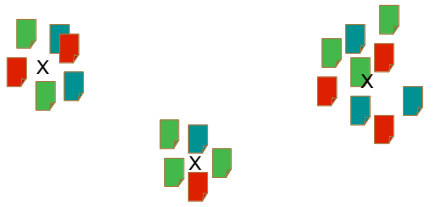
(*D* is a digit)

EM as Clustering

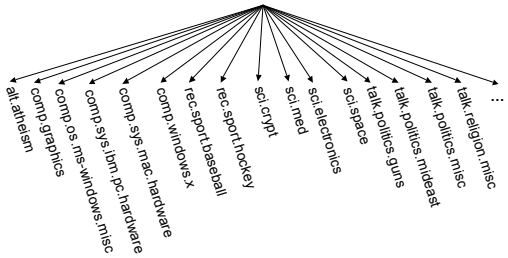


= unlabeled

EM as Clustering, Gone Wrong



20 Newsgroups Data Set



20 class labels, 20,000 documents
62k unique words

Newsgroups Classification Accuracy varying # labeled documents