CS 294-5: Statistical Natural Language Processing

Naïve-Bayes, Text Cat

Lecture 3: 9/12/05

Overview

- So far: language models give P(s)
  - Help model fluency for various noisy-channel processes (MT, ASR, etc.)
  - N-gram models don't represent any deep variables involved in language structure or meaning
  - Usually we want to know something about the input other than how likely it is (syntax, semantics, topic, etc)
- Next: Naïve-Bayes models
  - We introduce a single new global variable
  - Still a very simplistic model family
  - Lets us model hidden properties of text, but only very non-local ones…

Text Categorization

- Want to classify documents into broad semantic topics (e.g. politics, sports, etc.)

Democratic vice presidential candidate John Edwards on Sunday accused President Bush and Vice President Dick Cheney of misleading Americans by implying a link between deposed Iraqi President Saddam Hussein and the Sept. 11, 2001 terrorist attacks.

While No. 1 Southern California and No. 2 Oklahoma had no problems holding on to the top two spots with lopsided wins, four teams fell out of the rankings — Kansas State and Missouri from the Big 12 and Clemson from the Atlantic Coast Conference and Oregon from the Pac-10.

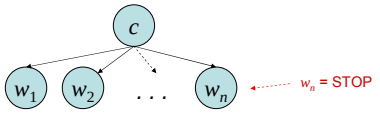
- Which one is the politics document? (And how much deep processing did that decision take?)
- One approach: bag-of-words and Naïve-Bayes models
- Another approach next lecture…

Naïve-Bayes Models

- Idea: pick a topic, then generate a document using a language model for that topic.
- Naïve-Bayes assumption: all words are independent given the topic.

$$P(c, w_1, w_2, \ldots w_n) = P(c)\prod_i P(w_i \mid c)$$

We have to smooth these!



- Compare to a unigram language model:

$$P(w_1, w_2, \ldots w_n) = \prod_i P(w_i)$$

Using NB for Classification

- We have a joint model of topics and documents

$$P(c, w_1, w_2, \ldots w_n) = P(c)\prod_i P(w_i \mid c)$$

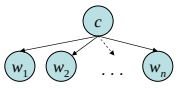
- Gives posterior likelihood of topic given a document

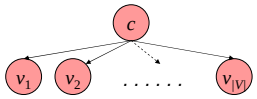
$$P(c \mid w_1, w_2, \ldots w_n) = \frac{P(c)\prod_i P(w_i \mid c)}{\sum_{c'}\left[P(c')\prod_i P(w_i \mid c')\right]}$$

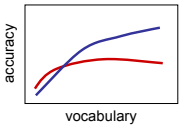
- What about totally unknown words?
- Can work shockingly well for textcat (especially in the wild)
- How can unigram models be so terrible for language modeling, but class-conditional unigram models work for textcat?
- Numerical / speed issues
- How about NB for spam detection?

Two NB Formulations

- Two NB models for text categorization
  - The class-conditional unigram model, a.k.a. multinomial model
    - One node per word in the document
    - Driven by words which are present
    - Multiple occurrences, multiple evidence
    - Better overall – plus, know how to smooth
  - The binary model
    - One node for each word in the vocabulary







- Incorporates explicit negative correlations
- Know how to do feature selection (e.g. keep words with high mutual information with the class variable)

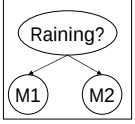
Example: Barometers

Reality

Raining | Sunny

P(+,+,r) = 3/8 | P(-,-,r) = 1/8 | P(+,+,s) = 1/8 | P(-,-,s) = 3/8

NB Model



NB FACTORS:

- P(s) = 1/2
- P(+|s) = 1/4
- P(+|r) = 3/4

PREDICTIONS:

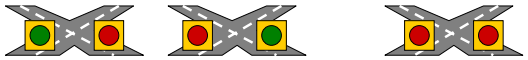
- P(r,+,+) = (½)(¾)(¾)
- P(s,+,+) = (½)(¼)(¼)
- P(r|+,+) = 9/10
- P(s|+,+) = 1/10

Overconfidence!

Example: Stoplights

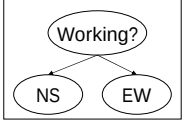
Reality

Lights Working | Lights Broken



P(g,r,w) = 3/7 | P(r,g,w) = 3/7 | P(r,r,b) = 1/7

NB Model



NB FACTORS:

- P(w) = 6/7
- P(r|w) = 1/2
- P(g|w) = 1/2
- P(b) = 1/7
- P(r|b) = 1
- P(g|b) = 0

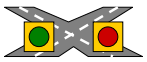
P(b|r,r) = 4/10 (what happened?)

(Non-)Independence Issues

- Mild Non-Independence
  - Evidence all points in the right direction
  - Observations just not entirely independent
  - Results
    - Inflated Confidence
    - Deflated Priors
  - What to do? Boost priors or attenuate evidence

$$P(c, w_1, w_2, \ldots w_n) \text{ "=" } P(c)^{boost>1} \prod_i P(w_i \mid c)^{boost<1}$$

- Severe Non-Independence
  - Words viewed independently are misleading
  - Interactions have to be modeled
  - What to do?
    - Change your model!



Language Identification

- How can we tell what language a document is in?

The 38th Parliament will meet on Monday, October 4, 2004, at 11:00 a.m. The first item of business will be the election of the Speaker of the House of Commons. Her Excellency the Governor General will open the First Session of the 38th Parliament on October 5, 2004, with a Speech from the Throne.

La 38e législature se réunira à 11 heures le lundi 4 octobre 2004, et la première affaire à l'ordre du jour sera l'élection du président de la Chambre des communes. Son Excellence la Gouverneure générale ouvrira la première session de la 38e législature avec un discours du Trône le mardi 5 octobre 2004.

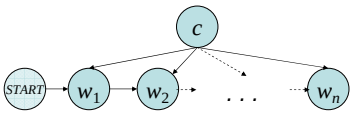
- How to tell the French from the English?
  - Treat it as word-level textcat?
    - Overkill, and requires a lot of training data
    - You don't actually need to know about words!

    Σύμφωνο σταθερότητας και ανάπτυξης
    Patto di stabilità e di crescita
  - Option: build a character-level language model

Class-Conditional LMs

- Can have a topic variable for other language models

$$P(c, w_1, w_2, \ldots w_n) = P(c) \prod_i P(w_i \mid w_{i-1}, c)$$



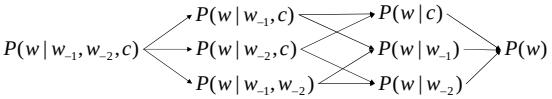
- Could be characters instead of words, used for language ID (HW2)
- Could sum out the topic variable and use as a language model
- How might a class-conditional n-gram language model behave differently from a standard n-gram model?

History Lattices

- Often we have multinomials which condition on lots of other events

$$P(w \mid w_{-1}, w_{-2}, c)$$

- Induce back-off lattices



- Can either pick a linear chain or use multiple mixing weights

$$P(w \mid w_{-1}, w_{-2}, c) \rightarrow P(w \mid w_{-1}, c) \rightarrow P(w \mid c) \rightarrow P(w)$$

- Often a sign that one should use other techniques, such as maximum entropy modeling (next class)

EM for Mixing Parameters

- How to estimate mixing parameters?

$$P_{LIN(\lambda_1,\lambda_2)}(w \mid w_{-1}) = \lambda_1 \hat{P}(w \mid w_{-1}) + \lambda_2 \hat{P}(w)$$

  - Sometimes you can just do line search
  - … or the "try a few orders of magnitude" approach
- Alternative: Use EM
  - Think of mixing as a hidden choice between histories:

$$P_{LIN(P_H)}(w \mid w_{-1}) = P_H(1)\hat{P}(w \mid w_{-1}) + P_H(0)\hat{P}(w)$$

  - Given a guess at P_H, we can calculate expectations of which generation route a given token took (over held-out data, why?)

$$P(h=1 \mid w, w_{-1}) = \frac{P_H(1)\hat{P}(w \mid w_{-1})}{P_H(1)\hat{P}(w \mid w_{-1}) + P_H(0)\hat{P}(w)}$$

  - Use these expectations to update P_H, rinse and repeat

EM for Naïve-Bayes

- First we calculate posteriors:

$$P(y|x) = \frac{P(y)\prod_i P(x_i|y)}{\sum_{y'} P(y')\prod_i P(x_i|y')}$$

- Then we re-estimate P(y), P(x|y) from the fractionally labeled data

$$c(x_i, y) = \sum_{(x,y)\in D} P(y|x)\,[c(x_i \in x)]$$

- Can do this when some or none of the docs are labeled

EM in General

- EM is a technique for learning when we have incomplete data (x,y)
  - Convenience Scenario: we want P(x), including y just makes the model simpler (e.g. mixing weights)
  - Induction Scenario: we actually want to know y (e.g. clustering)
- General approach: learn y and θ
  - E-step: make a guess at posteriors P(y|x,θ)
    - This means scoring all completions with the current parameters
  - M-step: fit θ to maximize P(x,y|θ)
    - This is usually the easy part – treat the completions as (fractional) complete data
  - We'll see lots of examples in this course
- EM is only locally optimal (why?)

Word Senses

- Words have multiple distinct meanings, or senses:
  - Plant: living plant, manufacturing plant, …
  - Title: name of a work, ownership document, form of address, material at the start of a film, …
- Many levels of sense distinctions
  - Homonymy: totally unrelated meanings (river bank, money bank)
  - Polysemy: related meanings (star in sky, star on tv)
  - Systematic polysemy: productive meaning extensions (organizations to their buildings) or metaphor
  - Sense distinctions can be extremely subtle (or not)
- Granularity of senses needed depends a lot on the task
- Why is it important to model word senses?
  - Translation, parsing, information retrieval?

Word Sense Disambiguation

- Example: living plant vs. manufacturing plant
- How do we tell these senses apart?
  - "context"

    The manufacturing plant which had previously sustained the town's economy shut down after an extended labor strike.

  - Maybe it's just text categorization
  - Each word sense represents a topic
  - Run the naive-bayes classifier from last class?
- Bag-of-words classification works ok for noun senses
  - 90% on classic, shockingly easy examples (line, interest, star)
  - 80% on senseval-1 nouns
  - 70% on senseval-1 verbs

Verb WSD

- Why are verbs harder?
  - Verbal senses less topical
  - More sensitive to structure, argument choice
- Verb Example: "Serve"
  - [function] The tree stump serves as a table
  - [enable] The scandal served to increase his popularity
  - [dish] We serve meals for the homeless
  - [enlist] He served his country
  - [jail] He served six years for embezzlement
  - [tennis] It was Agassi's turn to serve
  - [legal] He was served by the sheriff

Various Approaches to WSD

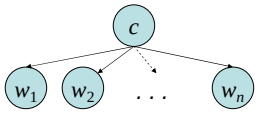
- Unsupervised learning
  - Bootstrapping (Yarowsky 95)
  - Clustering
- Indirect supervision
  - From thesauri
  - From WordNet
  - From parallel corpora
- Supervised learning
  - Most systems do some kind of supervised learning
  - Many competing classification technologies perform about the same (it's all about the knowledge sources you tap)
  - Problem: training data available for only a few words

Resources

- WordNet
  - Hand-build (but large) hierarchy of word senses
  - Basically a hierarchical thesaurus
- SensEval
  - A WSD competition, of which there have been 3 iterations
  - Training / test sets for a wide range of words, difficulties, and parts-of-speech
  - Bake-off where lots of labs tried lots of competing approaches
- SemCor
  - A big chunk of the Brown corpus annotated with WordNet senses
- OtherResources
  - The Open Mind Word Expert
  - Parallel texts
  - Flat thesauri

Knowledge Sources

- So what do we need to model to handle "serve"?
  - There are distant topical cues
    - …. point … court ……. serve …… game …



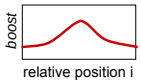
$$P(c, w_1, w_2, \ldots w_n) = P(c)\prod_i P(w_i \mid c)$$

Weighted Windows with NB

- Distance conditioning
  - Some words are important only when they are nearby
  - … as … point … court …… serve …… game …
  - …… serve as ……

$$P(c, w_{-k}, ..., w_{-1}, w_0, w_{+1}, \ldots w_{+k'}) = P(c)\prod_{i=-k}^{k'} P(w_i \mid c, bin(i))$$

- Distance weighting
  - Nearby words should get a larger vote
  - … court …… serve as…… game …… point



$$P(c, w_{-k}, ..., w_{-1}, w_0, w_{+1}, \ldots w_{+k'}) = P(c)\prod_{i=-k}^{k'} P(w_i \mid c)^{boost(i)}$$

Better Features

- There are smarter features:
  - Argument selectional preference:
    - serve NP[meals] vs. serve NP[papers] vs. serve NP[country]
  - Subcategorization:
    - [function] serve PP[as]
    - [enable] serve VP[to]
    - [tennis] serve <intransitive>
    - [food] serve NP {PP[to]}
  - Can capture poorly (but robustly) with local windows
  - … but we can also use a parser and get these features explicitly
- Other constraints (Yarowsky 95)
  - One-sense-per-discourse (only true for broad topical distinctions)
  - One-sense-per-collocation (pretty reliable when it kicks in: manufacturing plant, flowering plant)

Complex Features with NB?

- Example: Washington County jail served 11,166 meals last month - a figure that translates to feeding some 120 people three times daily for 31 days.
- So we have a decision to make based on a set of cues:
  - context:jail, context:county, context:feeding, …
  - local-context:jail, local-context:meals
  - subcat:NP, direct-object-head:meals
- Not clear how build a generative derivation for these:
  - Choose topic, then decide on having a transitive usage, then pick "meals" to be the object's head, then generate other words?
  - How about the words that appear in multiple features?
  - Hard to make this work (though maybe possible)
  - No real reason to try