Announcements

- Assignment 3
  - More Bayes nets
  - Due next week

- Reading Report #4
  - Will add to schedule later today

- Talk by Jordan Boyd-Graber today at 3pm
  - “Thinking on your Feet: Reinforcement Learning for Incremental Language Tasks”

Objectives

- Introduce the problem of text classification
- Introduce the Naïve Bayes model
- Classify with it!
- Deal with numerical precision issues

Text Classification

Goal: classify documents into broad semantic classes (e.g., sports, entertainment, technology, politics, 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 require?
  - Motivates an approach: bag-of-words, Naïve-Bayes models
  - Another approach in an upcoming lecture …

Generative Story

Idea: pick a class, then generate a document using a language model given that class.

What are the independence assumptions in this model?

Naïve Bayes

Model a global property of the text: the label c
- \( c \in \text{Var}(C) = \{c_1, c_2, \ldots, c_n\} \)
- Just includes one feature / random variable for each:
  - word token
  - word type in the vocabulary
  - \( P(c|w) \)
  - joint model of class label and document!

Dramatically naïve independence assumption:
- all words are conditionally independent of one another given the class.
Naïve-Bayes

- All words are conditionally independent of one another given the class.

\[ P(c, w_1, w_2, \ldots, w_n) = P(c) \prod_{i=1}^{n} P(w_i | c) \]

\[ \begin{align*}
P(c) & \quad P(w_j | c) \\
p(c) & \quad P(w_j | c) \\
p(c) & \quad P(w_j | c) \\
\cdots & \quad P(w_j | c) \\
\text{STOP} & \quad w_j \end{align*} \]

Posterior inference

- We can easily compute the posterior probability of a class given a document \( d = \{w_1, w_2, \ldots, w_n\} \).
- It’s just a conditional query on the model and can be rewritten with Bayes’ rule.

\[ P(c | d) = \frac{P(c, d)}{P(d)} = \frac{P(c)P(d | c)}{\sum_{c' \in \text{Range}(C)} P(c')P(d | c')} = \frac{P(c)\prod_{i} P(w_i | c)}{\sum_{c' \in \text{Range}(C)} P(c')\prod_{i} P(w_i | c')} \]

Classifying with Naïve Bayes

- Given document \( d \),

\[ P(c | d) = \frac{P(c)\prod_{i} P(w_i | c)}{\sum_{c' \in \text{Range}(C)} P(c')\prod_{i} P(w_i | c')} \]

\[ \hat{c} = \arg \max_{c} P(c | d) = \arg \max_{c} P(c)\prod_{i} P(w_i | c) = \arg \max_{c} P(c)\prod_{i} P(w_i | c) \]

Log(arithmetic) Domain

Photo credit: Nathan Davis and Aaron Davis, Spring 2007, Google Campus, Mountain View, CA

Classifying using Log Domain

\[ \hat{c} = \arg \max_{c} P(c | d) = \arg \max_{c} P(c)\prod_{i} P(w_i | c) \]

\[ = \arg \max_{c} \log P(c) + \sum_{i} \log P(w_i | c) \]

Practical Matters

- How easy is Naïve Bayes to train?
- How easy is it to test / predict / classify?
- What should we do with unknown words in a test document?
- Can work shockingly well for text classification (esp. in the wild). Why?
- How about NB for spam detection?

- Count frequencies of classes; count frequencies of word tokens with classes.
- Numerator of \( P(c | \text{doc}) \)
- Ignore them
- Isolated words can provide sufficient evidence.
- Definitely!
Proper Name Classification

- Movie: Beastie Boys: Live in Glasgow
- Person: Michelle Ford-Eriksson
- Place: Ramsbury
- Place: Market Bosworth
- Drug: Dilatab
- Drug: Cyanide Antidote Package
- Person: Bill Johnson
- Place: Ettalong
- Movie: The Suicide Club
- Place: Pézenas
- Company: AMLI Residential Properties Trust
- Drug: Diovan
- Place: Bucknell
- Movie: Marie, Nonna, la vierge et moi
- Person: Chevy Chase

Character-level evidence i.e., "features"

Insight

\[ \hat{c} = \arg \max_c P(c) \prod_j P(w_j | c) \]

- Think of these local model terms as a class-dependent "unigram model"

\[ \hat{c} = \arg \max_c P(c) \prod_j P(w_j) \]

Next

- More Naive Bayes