Bayes Nets Pt 3
Monday, March 26, 2018 8:57 AM

Last time

- we talked about text classification
  - task of predicting something about a text given only the text itself
    - Something: Author, genre, topic
  - Bag-of-words model to solve!
    - Probability and conditional probability are based solely on word frequency
    - Probability of class label depends on how often words are used
      - Can cause problems
        - Same word different meaning
        - abbreviations / stemming
        - Negation

- $C_{MAP} = \underset{j}{\text{argmax}}\ P(X_1, X_2, \ldots, X_n \mid C_j)\ P(C_j)$

$C_{NB} = \underset{j}{\text{argmax}}\ P(C_j) \prod_i^n P(X_1 \mid C_j)\ P(X_2 \mid C_j) \ldots P(X_n \mid C_j)$

Naive Bayes hypothesis

Assume X = "The quick brown fox"

$C_{NB} = \underset{j}{\text{argmax}}\ P(C_j) \prod P(X = \text{"The"} \mid C_j)\ P(X = \text{"quick"} \mid C_j)\ P(X = \text{"Brown"} \mid C_j)\ P(X = \text{"fox"} \mid C_j)$

How do we get these probabilities?

- Extract a vocabulary

- Every word, punctuation mark, or token in training set
  - 'n't 't / could + 'nt = couldn't

- Calculate $P(C_j)$
  - for all C_j calculate $\frac{|docs_j|}{|\#\text{ of documents}|}$ ← documents of class j
  - documents could large collections of text, such as books
  - Could be one line of dialog in a play
  - Initial guess at C distribution

- $P(X_k | C_j)$

  $Text_j$ = a single document containing all $docs_j$

  for each word in vocab, calculate

$$P(X_k | C_j) = \frac{n_k}{n}$$

  n_k ← number of times word k appears in $text_j$

  n ← # of words in $text_j$

That's all there is to it

- Unless...
- Underflow!
- Multiplying probabilities together results in VERY small numbers

- $\log(XY) = \log(X) + \log(Y)$

- Calculate log probability

$$C_{NB} = \underset{j}{\operatorname{argmax}} \log(P(c_j)) \sum_i \log(P(x_i \mid c_j))$$

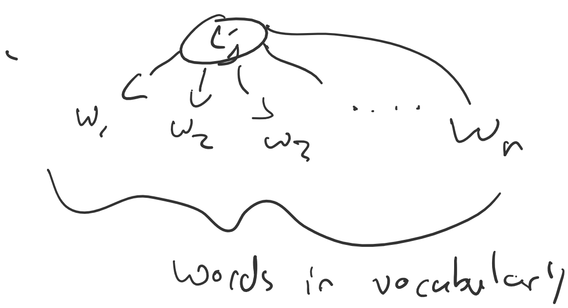
- The class with the highest log-score is still the most probable

- What if you haven't seen a word before?
  - doesn't exist in any class$_j$
    - Ignore it!
  - Doesn't exist in some class$_j$
    - Causes probabilities to vanish, regardless of other words in sentence
  - Solution: Pseudocounts

$$\hat{P}(x_i \mid C_j) = \frac{n_i + 1}{n + K}$$

\# of words in text$_j$ — \# of tokens in your sentence

  - Small probability associated with unseen words



- Naive Bayes, while powerful, is incredibly reductive.
- For more complex problems need a more complex network

- Tail to tail connections
  -

$X \to Y \to Z$

- X and Z are independent given Y
- even thouge X and Z are conditionally independent given Y, belief about X can propagate to Z