CMPT-882: Statistical Learning of Natural Language

Lecture #3

Anoop Sarkar
anoop@cs.sfu.ca
http://www.sfu.ca/~anoop

- Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval (1998). David Lewis. Proceedings of ECML-98 (10th meeting). pp. 4-15.

- A Comparison of Event Models for Naive Bayes Text Classification (1998). Andrew Mccallum and Kamal Nigam. In AAAI-98 Workshop on Learning for Text Categorization.

Document Classification

- Put a document into one of k classes.
- The only information available are the words in the document.
- We will look at the naive Bayes classifier as a framework for solving this task.

Bayes Rule

- C is a random variable over classes: $c_1, \ldots, c_k, \ldots, c_{e_C}$
- $\mathbf{X}$ is a random variable over a vector of attributes $\mathbf{x} = x_1, \ldots, x_j, \ldots, x_d$
- $P(C = c_k \mid \mathbf{X} = \mathbf{x}) = \frac{P(C{=}c_k) \times P(\mathbf{X}{=}\mathbf{x} | C{=}c_k)}{P(\mathbf{x})}$
- $P(c_k \mid \mathbf{x}) = \frac{P(c_k) \times P(\mathbf{x} | c_k)}{P(\mathbf{x})}$

Bayes Rule

- To prove Bayes rule: $P(A \mid B) = \frac{P(A) \times P(B|A)}{P(B)}$
- $P(A \mid B) = \frac{|A| \cap |B|}{|B|} = \frac{P(A,B)}{P(B)}$
- $P(A, B) = P(A \mid B) \times P(B) = P(B \mid A) \times P(A)$
- $P(A \mid B) = \frac{P(A) \times P(B|A)}{P(B)}$

Naive Bayes Assumption

- $P(c_k \mid \mathbf{x}) = \frac{P(c_k) \times P(\mathbf{x}|c_k)}{P(\mathbf{x})}$
- $P(\mathbf{x} \mid c_k) = \prod_{j=1}^{d} P(x_j \mid c_k)$
- $P(c_k \mid \mathbf{x}) = P(c_k) \times \prod_{j=1}^{d} P(x_j \mid c_k)$
- Class priors $P(c_k)$ need to be estimated: Laplace prior
  Each class gets the uniform distribution

Naive Bayes Assumption

- $P(c_k \mid \mathbf{x}) = P(c_k) \times P(\mathbf{x} \mid c_k)$
- θ is the set of parameter values for this model
- A particular setting of the values of these parameters defines a probability of the data
- $P(\mathbf{x} \mid \theta) = \sum_{k=1}^{e_C} P(c_k \mid \theta) \times \prod_{j=1}^{d} P(x_j \mid c_k; \theta)$

Naive Bayes Parameters

- $P(\mathbf{x} \mid \theta) = \sum_{k=1}^{eC} P(c_k \mid \theta) \times \prod_{j=1}^{d} P(x_j \mid c_k; \theta)$
- Maximum Likelihood Classifier (ML): $\widehat{\theta} = \underset{\theta}{\text{arg max}}\ P(\mathbf{x} \mid \theta)$
- Maximum A-Posteriori Classifier (MAP):
  $\widehat{\theta} = \underset{\theta}{\text{arg max}}\ P(\theta \mid \mathbf{x}) = \underset{\theta}{\text{arg max}}\ P(\mathbf{x} \mid \theta) \times P(\theta)$
  uses a prior over the parameter values
- Using the prior probability is a good idea. MAP classifiers perform better.

Text Representation for Document Classification

- The *bag of words* approach (word order information is lost)

- Two different event models within the bag of words approach:
  - *multi-variate Bernoulli* event model
    (also called Binary Independence Model)
  - *multinomial* event model

Sample Corpus

But other than the fact that besuboru is played with a ball and a bat , it ’s unrecognizable : Fans politely return foul balls to stadium ushers ; the strike zone expands depending on the size of the hitter ; ties are permitted -- even welcomed -- since they honorably sidestep the shame of defeat ; players must abide by strict rules of conduct even in their personal lives -- players for the Tokyo Giants , for example , must always wear ties when on the road .

Text Representation for Document Classification

- *multi-variate Bernoulli* event model
  $1, 0, 1, \ldots$

- *multinomial* event model
  $0, 3, 5, \ldots$
  typical smoothing step: add one to count of each word

Naive Bayes Classifier: multi-variate Bernoulli event model

- $\underset{c_k}{\text{arg max}}\ P(c_k \mid \mathbf{x}) = \underset{c_k}{\text{arg max}}\ P(c_k) \times P(\mathbf{x} \mid c_k)$

- Let the vocabulary V be represented as a vector for each document: $\mathbf{x} = w_1, \ldots, w_j, \ldots, w_d$

- d is the size of the vocabulary $\mid V \mid$

- $P(\mathbf{x} \mid c_k) = \prod_{k=1}^{|V|}(B_k P(w_k \mid c_k)) + (1 - B_k)(1 - P(w_k \mid c_k))$

Naive Bayes Classifier: multinomial event model

- $\underset{c_k}{\text{arg max}}\ P(c_k \mid \mathbf{x}) = \underset{c_k}{\text{arg max}}\ P(c_k) \times P(\mathbf{x} \mid c_k)$

- Let the vocabulary V be represented as a vector for each document: $\mathbf{x} = w_1, \ldots, w_j, \ldots, w_d$

- Let N_j be the frequency of word w_j in that document

- Let L be the length of the document

- $P(\mathbf{x} \mid c_k) = P(L) \times L! \times \prod_{j=1}^{|V|} \frac{P(w_j|c_k)^{N_j}}{N_j!}$

Feature selection

- Entropy of a random variable X where $P(X = x)$ is $H(X) = -\sum_{x \in X} P(x) \times \log P(x)$

- Mutual Information between two distributions is $I(X; Y) = H(X) - H(X \mid Y)$

- Feature selection using the mutual information between word (occurence) and the document class: $I(C; \mathbf{X})$

$$I(C; \mathbf{X}) = -\sum_{c_k \in C} P(c_k) log(P(c_k)) + \sum_{j} \sum_{c_k \in C} P(c_k \mid w_j) \times \log P(c_k \mid w_j)$$

Summary of Experimental Results

- Simple accuracy vs. recall/precision

$$\text{Recall} = \frac{\text{num of correct classes proposed}}{\text{num of classes in test data}}$$

$$\text{Precision} = \frac{\text{num of correct classes proposed}}{\text{num of classes proposed}}$$

- Multinomial event model always beats the Bernoulli event model (the most commonly used model)

- Issues with document length

- ifile: An Application of Machine Learning to E-mail Filtering (2000). Jason Rennie. Proceedings of the KDD-2000 Workshop on Text Mining.
  - Application of Naive Bayes to email filtering
  - Instance of online learning
  - Experiments with varying number of classes and number of documents (emails)

Homework: Use some Naive Bayes implementations

- `rainbow`: Naive Bayes classifiers for document classification based on the `bow` (bag of words) model by Andrew McCallum and collaborators.
  Use datasets from the CMU `textlearning` web page.

- Simple, generic Naive Bayes implementation by Christian Borgelt

Next Time

- Entropy and other concepts from Information Theory
  (Cover and Thomas, 1991)

- Moving beyond simple classification:
  Sequence Analysis with Hidden Markov Models