UNIVERSITY of DELAWARE UNIVERSITY of DE

Text Categorization – generative methods

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware

UNIVERSITY of DELAWARE UNIVERSITY of DE

Problem Setup of Text Categorization

- Input:
  - a document d
  - a fixed set of categories $C = \{c_1, c_2, \ldots, c_J\}$

A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$

a learned classifier: f(d) → c

- Output: a predicted category $c \in C$ for the document d

UNIVERSITY of DELAWARE UNIVERSITY of DE

Generative Classification Method

- Input:
  - a document d
  - a fixed set of categories $C = \{c_1, c_2, \ldots, c_J\}$
  - a training set of m labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
- Training
  - For each class, we estimate a probability model
- Testing
  - Given the document d, we select the category $c \in C$ which is most likely used to generate d.

UNIVERSITY of DELAWARE UNIVERSITY of DE

Naïve Bayes Classification

UNIVERSITY of DELAWARE UNIVERSITY of DE

Naïve Bayes Intuition

- A simple classification method based on Bayes rule
- Relies on very simple representation of document
  - Bag of words

UNIVERSITY of DELAWARE UNIVERSITY of DE

Text Classification with Naive Bayes

- Given a document d, we need to find c that is most likely used to generate d.

$$c^* = \arg\max_{c_i \in C} P(c_i \mid d)$$ **Posterior probability of c_i**

$$= \arg\max_{c_i \in C} \frac{P(d \mid c_i)P(c_i)}{P(d)}$$

$$= \arg\max_{c_i \in C} P(d \mid c_i)P(c_i)$$

Document model for category c_i **Prior probability of c_i**

How to estimate P(d|c)?

- Naïve assumption
  - A document is generated by a unigram language model.
  - Each word is generated *independently* of the others.

$$P(d \mid c_i) = P(w_1,...,w_n \mid c_i) = \prod_{j=1}^{n} P(w_j \mid c_i)$$

How to estimate it?

$$P(d \mid c_i) = P(w_1,...,w_n \mid c_i) = \prod_{j=1}^{n} P(w_j \mid c_i)$$

Fraction of times word w appears among all words in documents of topic c

$$P(w_j \mid c_i) = \frac{\sum_{k=1..m, class(d_k)=c_i} count(w_j, d_k)}{\sum_{w \in V} \sum_{k=1..m, class(d_k)=c_i} \text{count(w,d}_k)}$$

- Create a mega-document for topic c by concatenating all docs in this topic
- Use frequency of w in the mega-document.

$$P(d \mid c_i) = P(w_1,...,w_n \mid c_i) = \prod_{j=1}^{n} P(w_j \mid c_i)$$

Fraction of times word w appears among all words in documents of topic c

$$P(w_j \mid c_i) = \frac{\sum_{k=1..m, class(d_k)=c_i} count(w_j, d_k)}{\sum_{w \in V} \sum_{k=1..m, class(d_k)=c_i} \text{count(w,d}_k)}$$

This gives score of zero if d contains a unseen word

$$P(w_j \mid c_i) = \frac{\sum_{k=1..m, class(d_k)=c_i} count(w_j, d_k) + 1}{\sum_{w \in V} \sum_{k=1..m, class(d_k)=c_i} \text{count(w,d}_k) + |V|}$$

Text Classification with Naive Bayes

- Given a document d, we need to find c that is most likely used to generate d.

$$c^* = \arg\max_{c_i \in C} P(c_i \mid d)$$ Posterior probability of c_i

$$= \arg\max_{c_i \in C} \frac{P(d \mid c_i) P(c_i)}{P(d)}$$

$$= \arg\max_{c_i \in C} P(d \mid c_i) P(c_i)$$

Document model for category c_i | Prior probability of c_i

Implementation of Naïve Bayes

- Calculate $P(c)$
  For each c

$$P(c) \leftarrow \frac{\#of \text{ training documents with label c}}{\#of \text{ total training documents}}$$

- Calculate $P(w_i \mid c)$
  - *Text(c)* ←*concatenate all training documents with label c*
  - For each word w_i in V
    $n_i \leftarrow$ # of occurrences of w_i in *Text(c)*
  - $n \leftarrow$ # of all word occurrences in Text (c)
  - $P(w_i \mid c) \leftarrow \frac{n_i + 1}{n + |V|}$

An Example for Naïve Bayes Classification

- Problem:
  - Decide whether a document talks about China
- Training set
  - Doc 1: Chinese Beijing Chinese; Yes
  - Doc 2 : Chinese Chinese Shanghai; Yes
  - Doc 3: Chinese Macao; Yes
  - Doc 4: Tokyo Japan Chinese; No
- Test set
  - Doc 5: Chinese Chinese Chinese Tokyo Japan; ?

An Example (Cont.)

- Prior: p(c)=3/4 , p(not c) = ¼
- Conditional probabilities
  - P(Chinese|c) = (5+1)/(8+6) = 6/14=3/7
  - P(Tokyo|c) = P(Japan|c) = (0+1)/(8+6) = 1/14
  - P(Chinese | not c) = (1+1)/(3+6) = 2/9
  - P(Tokyo|not c) = P(Japan|not c)=(1+1)/(3+6)=2/9
- Posterior probabilities
  - P(c|d5) = 3/4 * (3/7)^3 * 1/14 * 1/14 =0.0003
  - P(not c |d5) = 1/4 * (2/9)^3 * 2/9 * 2/9 = 0.0001
- Yes. Doc5 talks about China.

Summary

- Pros:
  - Efficient
  - easy-to-implement
  - Very good in domains with many equally important features
- Cons:
  - Seldom gives the very best performance