Content-Based Recommendation (Part B)

Topic 5B

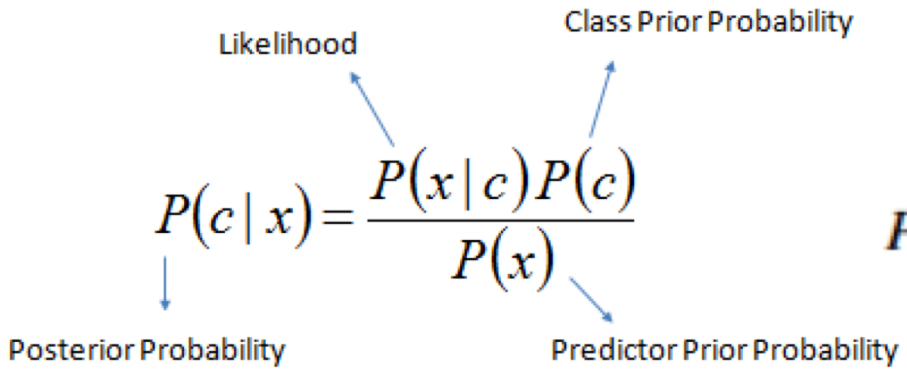
Other Text Classification Methods

- One way of deciding whether or not a document will be of interest to a user is to view the problem as a classification task, in which the possible cases are “like” and “dislike.”
- Classification Methods
  - Probabilistic methods
  - Other linear classifiers and machine learning
  - Explicit decision models
  - Feature selection

Probabilistic Methods

- Based on the naïve Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



$$P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Probabilistic Methods

- Based on the naïve Bayes Theorem:

Table 3.3. *Classification based on Boolean feature vector.*

Doc-ID	recommender	intelligent	learning	school	**Label**
1	1	1	1	0	**1**
2	0	0	1	1	**0**
3	1	1	0	0	**1**
4	1	0	1	1	**1**
5	0	0	0	1	**0**
6	1	1	0	0	**?**

$$P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

$$\begin{aligned} P(X|\text{Label}=1) = {} & P(\text{recommender}=1|\text{Label}=1) \times \\ & P(\text{intelligent}=1|\text{Label}=1) \times \\ & P(\text{learning}=0|\text{Label}=1) \times P(\text{school}=0|\text{Label}=1) \\ = {} & 3/3 \times 2/3 \times 1/3 \times 2/3 \\ \approx {} & 0.149 \end{aligned}$$

Probabilistic Methods

- In CF, the classifier based on the probabilistic techniques is commonly used to determine the membership of the active user in a cluster of users with similar preferences, whereas in content-based recommendation the classifier can also be directly used to determine the interestingness of a document.
- It has been found that the naïve Bayes model works well even though individual events are not conditionally independent (i.e., term occurrences are dependent)
- Needs a certain amount of training data for the technique to work.
- The Boolean feature vector approach does not reflect term counts.

Probabilistic Methods

Table 3.4. *Classification example with term counts.*

DocID	Words	Label
1	recommender intelligent recommender	1
2	recommender recommender learning	1
3	recommender school	1
4	teacher homework recommender	0
5	recommender recommender recommender teacher homework	?

$$P(v_i|C = c) = \frac{CountTerms(v_i, docs(c))}{AllTerms(docs(c))}$$

Returns the number of appearances of term v_i in documents labeled with c

Returns the number of all terms in these documents

Probabilistic Methods

Table 3.4. *Classification example with term counts.*

DocID	Words	Label
1	recommender intelligent recommender	1
2	recommender recommender learning	1
3	recommender school	1
4	teacher homework recommender	0
5	recommender recommender recommender teacher homework	?

- Laplace applied

$$\hat{P}(v_i|C = c) = \frac{CountTerms(v_i, docs(c)) + 1}{AllTerms(docs(c)) + |V|}$$

Number of different terms appearing in all documents (called the "vocabulary")

Probabilistic Methods

Table 3.4. *Classification example with term counts.*

DocID	Words	Label
1	recommender intelligent recommender	1
2	recommender recommender learning	1
3	recommender school	1
4	teacher homework recommender	0
5	recommender recommender recommender teacher homework	?

$$\hat{P}(recommender|Label = 1) = (5 + 1)/(8 + 6) = 6/14$$

$$\hat{P}(homework|Label = 1) = (0 + 1)/(8 + 6) = 1/14$$

$$\hat{P}(teacher|Label = 1) = (0 + 1)/(8 + 6) = 1/14$$

$$\hat{P}(recommender|Label = 0) = (1 + 1)/(3 + 6) = 2/9$$

$$\hat{P}(homework|Label = 0) = (1 + 1)/(3 + 6) = 2/9$$

$$\hat{P}(teacher|Label = 0) = (1 + 1)/(3 + 6) = 2/9$$

Probabilistic Methods

Table 3.4. *Classification example with term counts.*

DocID	Words	Label
1	recommender intelligent recommender	1
2	recommender recommender learning	1
3	recommender school	1
4	teacher homework recommender	0
5	recommender recommender recommender teacher homework	?

$$\hat{P}(Label = 1|v_1 \ldots v_n) = 3/4 \times (3/7)^3 \times 1/14 \times 1/14 \approx 0.0003$$

$$\hat{P}(Label = 0|v_1 \ldots v_n) = 1/4 \times (2/9)^3 \times 2/9 \times 2/9 \approx 0.0001$$

Other Linear Classifiers and Machine Learning

- Vector Space Representation
  - Each document is a vector, one component (term weight) for each term (= word).
  - Normally normalize vectors to unit length.
  - High-dimensional vector space:
    - Terms are axes
    - 10,000+ dimensions, or even 100,000+
    - Docs are vectors in this space
  - How can we do classification in this space?

Other Linear Classifiers and Machine Learning

- The training set is a set of documents, each labeled with its class (e.g., topic)
- In vector space based representation, this set corresponds to a labeled set of points (or, equivalently, vectors) in the vector space
  - Premise 1: Documents in the same class form a contiguous region of space
  - Premise 2: Documents from different classes don't overlap (much)
- We define surfaces to delineate classes in the space

Classification Using Vector Spaces

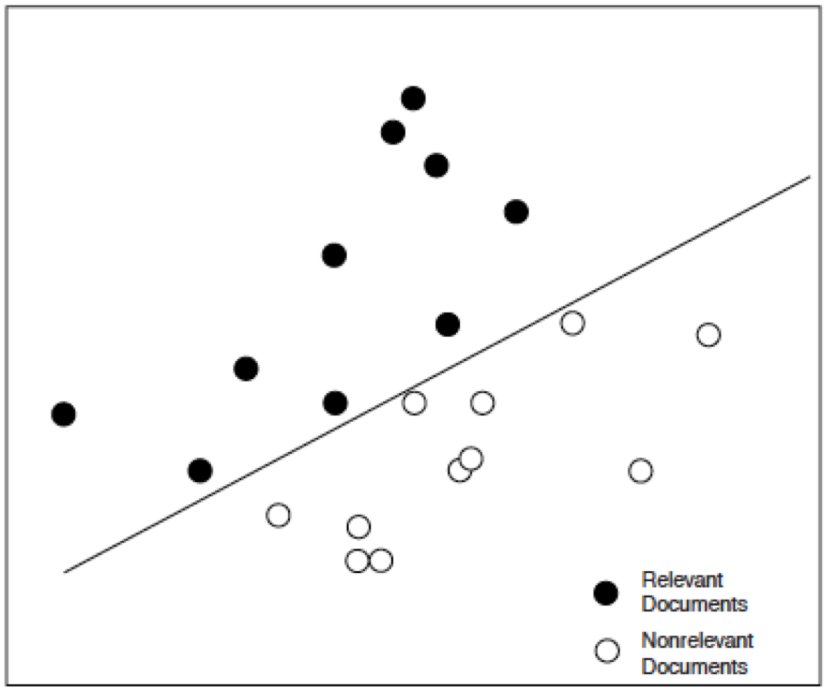


Figure 3.3. A linear classifier in two-dimensional space.

The line that we search for has the form $w_1x_1 + w_2x_2 = b$, where x_1 and x_2 correspond to the vector representation of a document and w_1, w_2, and b are the parameters to be learned.

In *n*-dimensional space, a generalized equation using weight and feature vectors:

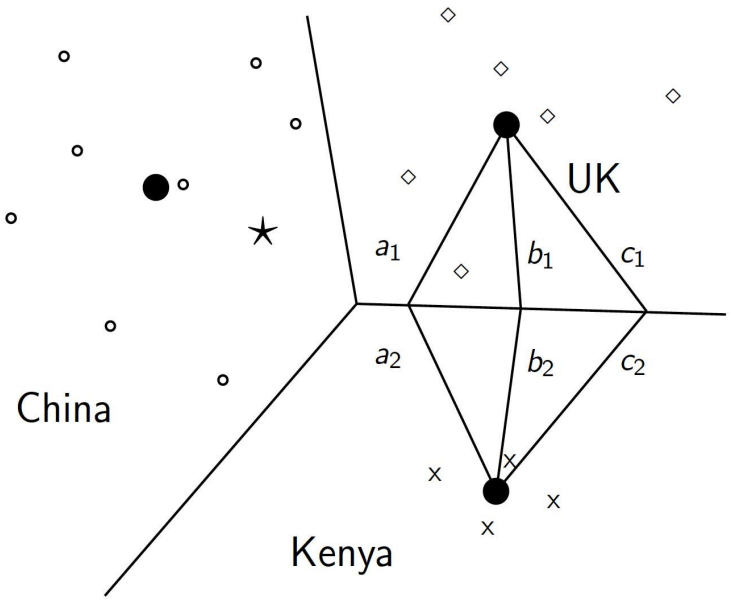
$$\vec{w}^T\vec{x} = b$$

Linear Classifiers

- Many common text classifiers are linear classifiers
  - Naïve Bayes
  - Perceptron
  - Rocchio
  - Logistic regression
  - Support vector machines
  - Linear regression

Rocchio Classification

- Rocchio forms a simple representation for each class: the centroid/prototype
- Classification is based on similarity to / distance from the prototype/centroid
- It is little used outside text classification, but has been used quite effectively for text classification



Explicit Decision Models

- As we have to work on relatively large feature sets in the content-based recommendation problem setting, decision trees are seldom used.
- Decision tree learners work best when a relatively small number of features exist.
- Decision trees can be used in recommender systems in combination with other techniques to improve recommendation efficiency or accuracy.

Feature Selection

- Document vectors tend to be very long and very sparse
- Feature selection is the process of choosing a subset of the available terms
  - Here, features mean terms
- Research shows that the recommendation accuracy improves when irrelevant features are removed.
  - Need to remove the words that are “too rare” or “too frequent”

Feature Selection

- Chi-square (χ^2) test can be used for feature selection.

Table 3.5. χ^2 *contingency table.*

	Term *t* appeared	Term *t* missing
Class "relevant"	A	B
Class "irrelevant"	C	D

$$\chi^2 = \frac{(A+B+C+D)(AD-BC)^2}{(A+B)(A+C)(B+D)(C+D)}$$

- Higher values indicate that the events of term occurrence and membership in a class not independent.
- To select features based on the test, the terms are first ranked.
- The optimal number of features are experimentally determined

Limitations

- Shallow content analysis
- Overspecialization
- Acquiring initial ratings