Modern Information Retrieval

Text classification: Naive Bayes [1]

Hamid Beigy

Sharif university of technology

November 20, 2022

[1] Some slides have been adapted from slides of Manning, Yannakoudakis, and Schütze.

Table of contents

Introduction

Introduction

1. How would you write a program that would automatically detect and delete this type of message?

```
From: ‘‘’’ <takworlld@hotmail.com>
Subject: real estate is the only way... gem  oalvgkay
Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even thousands for similar courses
I am 22 years old and I have already purchased 6 properties using the
methods outlined in this truly INCREDIBLE ebook.
Change your life NOW !
=================================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
=================================================
```

Examples of how search engines use classification

1. Query classification (types of queries)
2. Spelling correction
3. Document/webpage classification
4. Automatic detection of spam pages (spam vs. non-spam)
5. Topic classification (relevant to topic vs. not)
6. Language identification (classes: English vs. French etc.)
7. User classification (personalised search)

Classification terminology

1. A document space $\mathbb{X}$

   Documents are represented in this space (typically some type of high-dimensional space).
2. A fixed set of classes $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$

   The classes are human-defined for the needs of an application (e.g., spam vs. nonspam).
3. A training set $\mathbb{D}$ of labeled documents.

   Each labeled document $\langle d, c\rangle \in \mathbb{X} \times \mathbb{C}$
4. Using a learning method or learning algorithm, we then wish to learn a classifier γ that maps documents to classes:

$$\gamma : \mathbb{X} \rightarrow \mathbb{C}$$

5. Classification task:

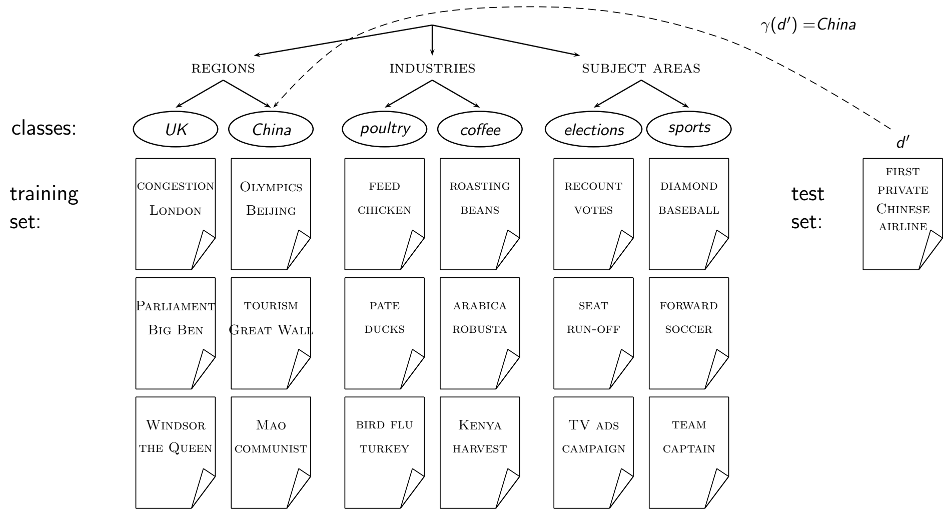
   Given a description $d \in \mathbb{X}$ of a document, determine $\gamma(d) \in \mathbb{C}$, that is, the class that is most appropriate for d.

Classification terminology

1. Features: measurable properties of the data.
2. Classes: labels associated with the data.
3. Consider the following example
   - Sentiment classification: automatically classify text based on the sentiment it contains (e.g., movie reviews).
   - Features: the words the text contains, parts of speech, grammatical constructions etc.
   - Classes: positive or negative sentiment (binary classification).
4. Classification is the function that maps input features to a class.

Classification terminology

1. Consider a text classification with six classes {UK, China, poultry, coffee, elections, sports}



Classification methods

Classification methods: Manual

1. Manual classification was used by Yahoo in the beginning of the web. Also: PubMed
2. Very accurate if job is done by experts
3. Consistent when the problem size and team is small
4. Scaling manual classification is difficult and expensive.
5. Hence, we need automatic methods for classification.

Classification methods: Rule-based

1. E.g., Google Alerts is rule-based classification.
2. There are IDE-type development environments for writing very complex rules efficiently. (e.g., Verity)
3. Often: Boolean combinations (as in Google Alerts)
4. Accuracy is very high if a rule has been carefully refined over time by a subject expert.
5. Building and maintaining rule-based classification systems is cumbersome and expensive.
6. Email classification in email clients such as outlook.

Classification methods: Statistical/Probabilistic

1. This was our definition of the classification problem – text classification as a learning problem
   - Supervised learning of a the classification function γ and
   - application of γ to classifying new documents
2. We will look at two methods for doing this: Naive Bayes and SVMs
3. No free lunch: requires hand-classified training data
4. But this manual classification can be done by non-experts.

Naive Bayes classifier

Bayes classifier

1. We compute the probability of a document d being in a class c as follows:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

$$P(c|d) \propto P(c)P(d|c)$$

2. $P(d)$ is constant during a given classification and won't affect the result.

Naive Bayes classifier

1. The Naive Bayes classifier is a probabilistic classifier.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

2. n_d is the length of the document. (number of tokens)
3. $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c
4. $P(t_k|c)$ as a measure of how much evidence t_k contributes that c is the correct class.
5. $P(c)$ is the prior probability of c.
6. If a document's terms do not provide clear evidence for one class vs. another, we choose the c with highest $P(c)$.

Maximum a posteriori class

1. Our goal in Naive Bayes classification is to find the "best" class.
2. The best class is the most likely or maximum a posteriori (MAP) class c_{map}:

$$c_{\text{map}} = \underset{c \in \mathbb{C}}{\operatorname{argmax}} \hat{P}(c|d) = \underset{c \in \mathbb{C}}{\operatorname{argmax}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)$$

Maximum a posteriori class

1. Multiplying lots of small probabilities can result in floating point underflow.
2. Since $\log(xy) = \log(x) + \log(y)$, we can sum log probabilities instead of multiplying probabilities.
3. Since log is a monotonic function, the class with the highest score does not change.
4. So what we usually compute in practice is:

$$c_{\text{map}} = \underset{c \in \mathbb{C}}{\text{argmax}} \; [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)]$$

Naive Bayes classifier

1. Classification rule:

$$c_{\text{map}} = \underset{c \in \mathbb{C}}{\operatorname{argmax}} \left[\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k | c) \right]$$

2. Simple interpretation:
    - Each conditional parameter $\log \hat{P}(t_k|c)$ is a weight that indicates how good an indicator t_k is for c.
    - The prior $\log \hat{P}(c)$ is a weight that indicates the relative frequency of c.
    - The sum of log prior and term weights is then a measure of how much evidence there is for the document being in the class.
    - We select the class with the most evidence.

Parameter estimation

1. Estimate parameters $\hat{P}(c)$ and $\hat{P}(t_k|c)$ from train data: How?
2. Prior:

$$\hat{P}(c) = \frac{N_c}{N}$$

   - N_c is the number of docs in class c;
   - N is the total number of docs.
3. Conditional probabilities:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

   T_{ct} is the number of tokens of t in training documents from class c (includes multiple occurrences)
4. We've made a Naive Bayes independence assumption here.

Maximum likelihood

1. Is it possible to compute $\hat{P}(c)$? (why?)
2. In some cases, we consider $\hat{P}(c)$ to be equal for all documents.
3. Hence, we only compute $\hat{P}(t|c)$
4. Conditional probabilities:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

   T_{ct} is the number of tokens of t in training documents from class c (includes multiple occurrences)
5. We've made a Naive Bayes independence assumption here.
6. Classification rule:

$$c_{\text{map}} = \operatorname*{argmax}_{c \in \mathbb{C}} \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)$$

Add-one smoothing

1. Before:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

2. Now: Add one to each count to avoid zeros:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

B is the number of bins – in this case the number of different words or the size of the vocabulary $|V| = M$

Time complexity of Naive Bayes

1. Time complexity is

mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{\text{ave}} + \lvert\mathbb{C}\rvert\lvert V\rvert)$
testing	$\Theta(L_{\text{a}} + \lvert\mathbb{C}\rvert M_{\text{a}}) = \Theta(\lvert\mathbb{C}\rvert M_{\text{a}})$

   - L_{ave}: average length of a training doc,
   - L_{a}: length of the test doc,
   - M_{a}: the number of distinct terms in the test doc,
   - $\mathbb{D}$: training set,
   - V: vocabulary,
   - $\mathbb{C}$: set of classes
2. $\Theta(\lvert\mathbb{D}\rvert L_{\text{ave}})$ is the time it takes to compute all counts.
3. $\Theta(\lvert\mathbb{C}\rvert\lvert V\rvert)$ is the time it takes to compute the parameters from the counts.
4. Generally: $\lvert\mathbb{C}\rvert\lvert V\rvert < \lvert\mathbb{D}\rvert L_{\text{ave}}$
5. Test time is also linear (in the length of the test document).
6. Thus: Naive Bayes is linear in the size of the training set (training) and the test document (testing). This is optimal.

An example

1. Consider the following dataset.

	docID	words in document	$c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

2. we have

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

B is the size of the vocabulary $|V| = M$.

$$c_{\text{map}} = \underset{c \in \mathbb{C}}{\text{argmax}} \left[\hat{P}(c) \cdot \prod_{1 \le k \le n_d} \hat{P}(t_k|c)\right]$$

An example: training

1. Priors: $\hat{P}(c) = \frac{3}{4}$ and $\hat{P}(\overline{c}) = \frac{1}{4}$ Conditional probabilities:

$$\begin{aligned} \hat{P}(\textsc{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\ \hat{P}(\textsc{Tokyo}|c) = \hat{P}(\textsc{Japan}|c) &= (0+1)/(8+6) = 1/14 \\ \hat{P}(\textsc{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\ \hat{P}(\textsc{Tokyo}|\overline{c}) = &= (1+1)/(3+6) = 2/9 \\ \hat{P}(\textsc{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9 \end{aligned}$$

2. The denominators are $(8+6)$ and $(3+6)$ because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B is 6 as the vocabulary consists of six terms.

An example: classification

1. For classification, we have

$$\begin{aligned}\hat{P}(c|d_5) &\propto \frac{3}{4} \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003 \\ \hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001\end{aligned}$$

2. Thus, the classifier assigns the test document to $c = China$.
3. The reason for this classification decision is that the three occurrences of the positive indicator Chinese in d_5 outweigh the occurrences of the two negative indicators Japan and Tokyo.

Performance measures of classifiers

Evaluating classification

1. Evaluation must be done on test data that are independent of the training data, i.e., training and test sets are disjoint.
2. It's easy to get good performance on a test set that was available to the learner during training (e.g., just memorize the test set).
3. Measures: Precision, recall, F_1, classification accuracy

Evaluating classification

1. Precision, recall, and F_1 can be calculated using

	in the class	not in the class
predicted to be in the class	true positives (TP)	false positives (FP)
predicted to not be in the class	false negatives (FN)	true negatives (TN)

2. TP, FP, FN, TN are counts of documents. The sum of these four counts is the total number of documents.
   - Precision: $P = TP/(TP + FP)$
   - Recall: $R = TP/(TP + FN)$
3. F_1 allows us to trade off precision against recall.

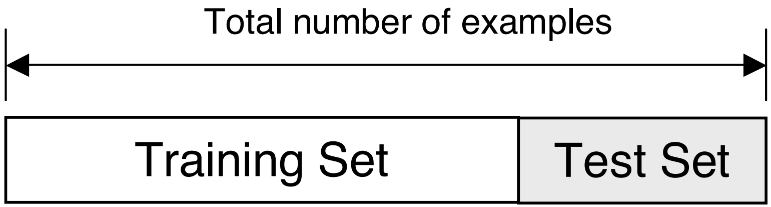
$$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P + R}$$

4. This is the harmonic mean of P and R: $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$

Evaluating the performance of a classifier

Hold-out method

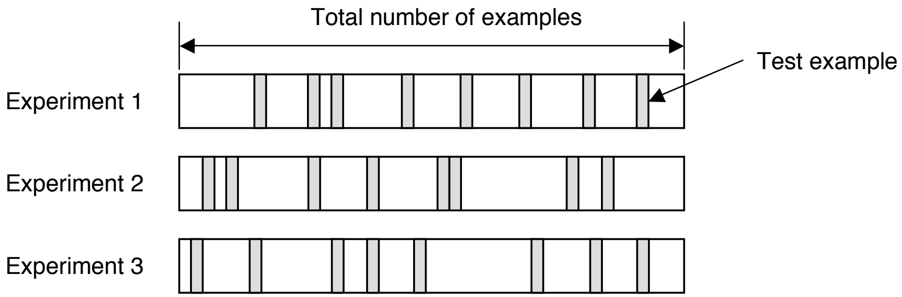
1. Hold-out Hold-out partitions the given data into two independent sets : training and test sets.



- Typically two-thirds of the data are allocated to the training set and the remaining one-third is allocated to the test set.
- The training set is used to drive the model.
- The test set is used to estimate the accuracy of the model.

Random sub-sampling method

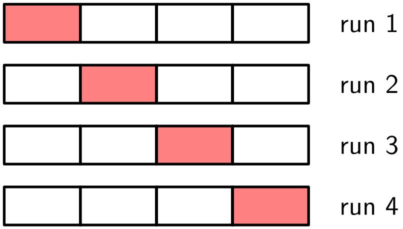
1. Random sub-sampling Random sub-sampling is a variation of the hold-out method in which hold-out is repeated k times.



- The estimated error rate is the average of the error rates for classifiers derived for the independently and randomly generated test partitions.
- Random sub-sampling can produce better error estimates than a single train-and-test partition (hold-out method).

Cross validation method

1. *K*-fold cross validation The initial data are randomly partitioned into K mutually exclusive subsets or folds, $S_1, S_2, \dots, S_K$, each of approximately equal size. .



- Training and testing is performed K times.
- In iteration k, partition S_k is used for test and the remaining partitions collectively used for training.
- The accuracy is the percentage of the total number of correctly classified test examples.
- The advantage of K-fold cross validation is that all the examples in the dataset are eventually used for both training and testing.

References

Reading

1. Chapter 13 of Information Retrieval Book[2]

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

References

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Questions?