Statistical Methods in Artificial Intelligence

Monsoon 2016

News Classification

Member 1:
Murali Krishna Reddy
201402078

Member 2:
Battu Varshit
201402029

November 3, 2016

Abstract

Text classification is the task of classifying documents by their content i.e , by the words of which they are comprised of. News Classification is an important task in the field of Language Processing. In this report news classification is done using two methods in Machine Learning, Naive Bayes and Random Forests. Classifiers are first trained on the training data and then tested on new data.

1 Introduction

The two features used for classification using Naive Bayes and Random Forests are Bernoulli and Multinomial document model.

Bernoulli document model: a document is represented by a feature vector with binary elements taking value 1 if the corresponding word is present in the document and 0 if the word is not present.

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \propto P(D|C)P(C)$$

Example: Let V = {blue, red, dog, cat, biscuit, apple}
document1 = [1, 0, 1, 0, 1, 0]T
document2 = [2, 0, 1, 0, 1, 0]T

As mentioned above, in the Bernoulli model a document is represented by a binary vector, which represents a point in the space of words. If we have a vocabulary V containing a set of |V| words, then the t^{th} dimension of a document vector corresponds to word wt in the vocabulary. Let bi be the feature vector for the i^{th} document D_i; then the t^{th} element of b_i, written bit, is either 0 or 1 representing the absence or presence of word wt in the i th document. Let P(wt|C) be the probability of word wt occurring in a document of class C; the probability of wt not occurring in a document of this class is given by (1 − P(wt|C)). If we make the naive Bayes assumption, that the probability of each word occurring in the document is independent of the occurrences of the other words, then we can write the document likelihood P(Di| C) in terms of the individual word likelihoods P(wt|C)

$$P(D_i|C) \sim P(b_i|C) = \prod_{t=i}^{|V|}[b_{it}P(w_t|C) + (1 - b_{it})(1 - P(w_t|C))]$$

This product goes over all words in the vocabulary. If word wt is present, then bit =1 and the required probability is P(wt|C); if word wt is not present, then bit =0 and the required probability is (1 − P(wt|C)). We can imagine this as a model for generating document feature vectors of class C, in which the document feature vector is modelled as a collection of |V| weighted coin tosses, the t^{th} having a probability of success equal to P(wt|C).

Multinomial document model: a document is represented by a feature vector with integer elements whose value is the frequency of that word in the document.

In the multinomial document model, the document feature vectors capture the frequency of words, not just their presence or absence. Let xi be the multinomial model feature vector for the i^{th} document D_i. The t^{th} element of x_i, written x_{it}, is the count of the number of times word w_t occurs in document D_i.

Let $n_i = \sum_t x_{it}$ be the total number of words in document Di.
Let P(wt|C) again be the probability of word wt occurring in class C.

This time estimated using the word frequency information from the document feature vectors. We again make the assumption, that the probability of each word occurring in the document is independent of the occurrences of the other words. We can then write the document likelihood P(Di|C) as a multinomial distribution, where the number of draws corresponds to the length of the document, and the proportion of drawing item t is the probability of word type t occurring in a document of class C, $P(w_t|C)$ of drawing item t is the probability of word type t occurring in a document of class C, $P(w_t|C)$.

$$P(D_i|C) \sim P(x_i|C) = \frac{n!}{\prod_{t=1}^{|V|} x_{it}!} \prod P(w_t|C)^{x_{it}} \propto \prod_{t=1}^{|V|} P(w_t|C)^{x_{it}}$$

Random Forest Model: This uses decision trees to classify the input. We went till a depth of 10 or till a node whose entropy is zero. We randomly split the available data into training and testing sets. The decision trees are built by selecting a fraction of the training data. We give the test data to every decision tree and classify them by assigning the class with the highest frequency.

Decision Tree: A predictive model that uses a set of binary rules applied to calculate a target value. It can be used for classification (categorical variables) or regression (continuous variables) applications. Rules are developed using software available in many statistics packages. Different algorithms are used to determine the "best" split at a node.

A different subset of the training data are selected (~2/3), with replacement, to train each tree. Remaining training data(OOB) are used to estimate error and variable importance. Class assignment is made by the number of votes from all of the trees and for regression the average of the results is used. A randomly selected subset of variables is used to split each node. The number of variables used is decided by the user (mtry parameter in R). Smaller subset produces less correlation (lower error rate) but lower predictive power (high error rate). Optimum range of values is often quite wide.

2 Results

The Naive Bayes with Multivariate Bernoulli achieved an accuracy of ~75%.

Table	Politics	Sports	Tech	Business	Entertainment	Accuracy
Politics	45	0	1	3	1	0.90
Sport	13	37	0	0	0	0.74
Tech	11	0	39	0	0	0.78
Business	14	0	5	31	0	0.62
Entertainment	13	0	1	1	35	0.70

The Naive Bayes with Normalised Multinomial achieved an accuracy of ~60%.

Table	Politics	Sports	Tech	Business	Entertainment	Accuracy
Politics	47	1	0	2	0	0.94
Sport	16	33	0	0	1	0.66
Tech	47	0	0	3	0	0.06
Business	24	0	1	25	0	0.50
Entertainment	0	0	1	39	35	0.78

The Random Forest with Multivariate Bernoulli achieved an accuracy of ~52%.

Table	Politics	Sports	Tech	Business	Entertainment	Accuracy
Politics	18	28	0	4	0	0.36
Sport	0	50	0	0	0	1
Tech	0	22	24	4	10	0.48
Business	0	20	0	30	0	0.60
Entertainment	0	40	0	2	8	0.16

The Random Forest with Normalised Multinomial achieved an accuracy of ~56%.

Table	Politics	Sports	Tech	Business	Entertainment	Accuracy
Politics	19	30	0	1	0	0.38
Sport	0	50	0	0	0	1
Tech	0	22	28	0	0	0.56
Business	0	13	0	37	0	0.74
Entertainment	1	42	0	1	6	0.12

3 Conclusion

In this chapter we have shown how the Naive Bayes approximation and Random Forest can be used for document classification, by constructing distributions over words. The classifiers require a document model to estimate P(document | class). We looked at two document models that we can use

- **Bernoulli document model** : A document is represented by a binary feature vector, whose elements indicate absence or presence of corresponding word in the document.
- **Multinomial document model** : a document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the document.

As of now the size of vocabulary is 2000 words. To achieve even better results we need to be using very large datasets which will be needing more time to classify.