A Naive Bayes implementation for classifying news headlines text

Bakthavatsalam, Tejas Mahale, Sachin Hagaribommanahalli
Principles of Artificial Intelligence

Abstract

A text classification algorithm is used to classify the incoming headlines into 4 broad categories:Business, Health, Entertainment, and Technology. The Naive Bayes is a popular technique for classifying text due to its relative simplicity and ease of implementation. In this project, we design a Naive Bayes classifier for classifying headlines based on only the text contained in them. We use the news aggregator dataset from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/News+Aggregator)

Introduction

Text classification is an intersting field of machine learning and its applications are diverse. Many algorithms exist for classifying text. However, a Naive Bayes classifier is a simple and effective method for classifying text.

The first step in the classifier design is feature selection. For text classification, a bag of words model is usually used. These features may be very large in numebr and some of these may not be benefcial. Further, nosiy features may affect the accuracy and speed. Feature selection is a nnecessary pre-processing step to reduce the dimensionality of the feature space and improve the effeciency of the classifier.

Once the features are selected, the classifier is trained and parameters of the model are estimated. These parameters are,

$$\phi_{i|y} = P(X_i = 1|Y = y) \tag{1}$$

$$\phi_y = P(Y = y) \tag{2}$$

and

$$y \in \{1, 2, 3, 4\}$$

This is the conditional probability mass function of the random variable X_i given Y where X_i represents the occurence of a word and Y_j represents the class it belongs to. X_i are assumed to be Bernoulli distributed.

After the parameters have been estiamted, an input document that is to be classified is fed to the classifier as a bag-of-words

$$X = [X_1 X_2 X_3 ... X_n] \tag{3}$$

where X_1, X_2,...,X_n represent the occurence of the word in the document. The classification is done according to Bayes rule and using the conditional independence given class assumption (Naive assumption).

Naive Bayes

Naive Bayes is a simple probabilistic classifier which exploits the Bayes theorem. A very strong assumption that the Naive Bayes makes is independece of features given a class label. Mathematically,

$$\begin{aligned} P(X_1 = x_1, .., X_n = x_n|Y = y) = \\ P(X_1 = x_1|Y = y)...P(X_n = x_n|Y = y) \end{aligned} \tag{4}$$

Given a documnet,

$$X = [X_1 X_2 X_3 ... X_n]$$

The classifier has to tag it with the appropriate class using the conditional probability of the class given document. That is, the classification is as follows:

The condtional probability for class Y=y given X=x is computed for all classes

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)} \tag{5}$$

Next, the the headline is assigned to the class with this highest conditional probability.

$$\begin{aligned} Y &= \arg\max_y P(Y = y|X = x) \\ &= \arg\max_y \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)} \end{aligned} \tag{6}$$

Since P(X=x) is the same for all cases, it can be ignored during the classification process. This gives,

$$Y = \arg\max_y P(X = x|Y = y)P(Y = y) \tag{7}$$

Because of the conditional independence given class, we use (3) in (6) and this reduces to,

$$Y = \arg\max_y \prod_{i=1}^{n} (P(X_i = x_i|Y = y))P(Y = y) \quad (8)$$

This is the required equation for classification.

Feature selection

The feature selection approach attempts to select the best *K* out of *N* terms to form the vocabulary[4]. A pre-processing step consisting of stop-word removal and stemming were also carried out to reduce the feature size. Document Frequency thresholding (DF) has proven to be one of the best feature selection method text classification. Document frequency signifies the occurrence of particular term in whole collection of documents and by setting threshold, terms occurred multiple times are retained. Terms which are non informative for classification can be removed. By using this, tens of hundreds rare features can be removed before the step of feature selection[4]. As document frequency can be calculated in entire test set, selecting features based on repeated word increases probability of these feature can be present in future test cases. Text domain has number of features and most of these features are not useful for text classification. In such case document frequency plays important role and speed up the categorisation process. Due to its simplicity and effectiveness, Document Frequency is adopted in more and more text mining experiments[1][2][3].

For the purpose of this project, we select the top 2000 words with most frequency for each class and combine them to use as the dictionary.

Classifier design

Once the bag-of-words features are selected, the conditional probability of each words given the class lablel has to be computed. That is, the parameters of the model needs to be estimated. The random variable here is asssumed to bernoulli distributed as the number of words in the headline is not of much importance. Just the presence of certain keywords will be used in the classification. Since we have assumed the random variables to be bernoulli distributed, the parameter is just the sample mean of the random varible.

$$\phi_{i|y} = \frac{\sum_{i=1}^{m} I\{X_i = 1, Y = y\}}{\sum_{i=1}^{m} I\{Y = y\}} \quad (9)$$

$$\phi_y = \frac{\sum_{i=1}^{m} I\{Y = y\}}{m} \quad (10)$$

where *m* is the number of training examples. (9) and (10) give the required estiamte of the probabilities for all words in the dictionary given the class. The *I* used here is the indicator random varibale. To calculate (9), we count the number of documents of class *y* which contains the word *i* and divide it by the number of documents of class *y*. Similarly, to calculate (10), we count the number of documents of class *y* and divide it by the total number of documents.
After the parameters have been estimated, (8) can be used to classify the headline. However, (8) is the product of many conditional probabilities and their values might be small. This can cause floating-point underflow. To overcome this issue, we use the logarithm probability. The class with the highest log probability score is our required classification. The classification rule is therefore,

$$Y = \arg\max_y \left\{ \sum_{i=1}^{n} \log(P(X_i = x_i|Y = y)) + \log(P(Y = y)) \right\} \quad (11)$$

(11) can be used provided the words in the document are in the dictionary. For documents that contain words not present in the dictioanry, the conditional probability is computed at runtime using the laplace smoothing method. This value can be approximated to,

$$\phi_{i|y} = \frac{1}{\sum_{i=1}^{m} I\{Y = y\} + m} \quad (12)$$

Performance metric

Any classifier developed has to have a general metric to measure its performance and evaluate the model against them for it to be used for real world data. A generic performance measure called Multiclass Performance Measure(MPS) is developed for pattern recognition [5]. For this multi-class classifier where the input is classified into one class out of given number of classes, different performance measures such as accuracy, precision and recall are used. Average accuracy, precision and recall are calculated as given in the Table 2 which uses the concept of confusion matrix for binary classification as a micro- or macro- average of Table 1 [6].

Table 1: Confusion Matrix for binary classification

Data class	Classified as positive	Classified as negative
Positive	True positive (tp)	True negative(tn)
Negative	False positive (fp)	False negative(fn)

Table 2: Measures for multi-class classification based on a generalization of the measures of Table 1 for many classes $C_i : tp_i$ are true positive for C_i , and fp_i false positive, fn_i false negative, and tn_i true negative counts respectively. l and M indices represent micro- and macro-averaging.

Measure	Formula
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{(tp_i+tn_i)}{(tp_i+fn_i+fp_i+tn_i)}}{l}$
Precision	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$
Recall	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$

Table 3: Confusion Matrix of the trained Naive Bayes model

Data Class	Business	Entertainment	Technology	Medical
Business	102237	3828	8967	935
Entertainment	2784	145773	3312	724
Technology	8578	4258	95021	646
Medical	3657	3596	1843	35841

Table 4: Measures for multi-class classification based on table 2 and table 3

Measure	Value (in percentage)
Average Accuracy	90.419
Precision	88.23
Recall	90.8

Results and Conclusion

The classifier performs well despite the naive Bayes assumption.A classification accuracy of 89.65% was achieved as shown in table 4. Precision of 88.02% signifies that the fraction of retrieved instances that are relevant is really high. Recall or sensitivity of 89.1% signifies that the algorithm returned the instances which are most relevant.

Bibliography

[1] Xu Y., Wang B., Li J., Jing H. (2008) An Extended Document Frequency Metric for Feature Selection in Text Categorization. In: Li H., Liu T., Ma WY., Sakai T., Wong KF., Zhou G. (eds) Information Retrieval Technology. AIRS 2008. Lecture Notes in Computer Science, vol 4993. Springer, Berlin, Heidelberg

[2] Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of CIKM 1998, pp. 148 155 (1998)

[3] Li, Y.H., Jain, A.K.: Classification of text documents. Comput. J. 41(8), 537 546 (1998)

[4] B. Tang, S. Kay, H. He, "Toward Optimal Feature Selection in Naive Bayes for Text Categorization", IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 9, pp. 2508-2521, 2016

[5] Thomas Kautz, Bjoern M. Eskofire, Cristian F. Pasluosta, Generic performance measure for multiclass classifier, In Pattern Recognition, Volume 68, 2017, Pages 111-125

[6] Marina Sokovola, Guy Lapalme, A Systematic analysis of performance measures for classification tasks, In Information Processing and Management, volume 45, Issue 4, 2009, pages 427-437