Journal Title
XX(X):1–4
©The Author(s) 0000
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

SuperBayesCat - A pairwise extention to the Naive Bayes Classifier

Dr Barry D.O. Adams[1]

Abstract

We describe an extension to the Naive Bayes Classifier, in which remove the assumption that all document features (words or ngrams) are independent, based on the computing the pairwise correlation of words (ngram) in the training documents, and find formulae for the training and test phases. A python implementation is available on github [?] and tests are run using the Reutes21578 accuracy increases from 79.1% to 81.1%, when the classifier also included features pair of word adjacent to each other. We discuss the speed and accuracy of the implementation and its advantages and disadvantages.

Keywords

bayes classifier document classification supervised

Introduction

The Naive Bayes Classifier is a subset of Bayesian decision theory and can classify documents surprisingly well [1].

In a naive Bayes a the probability of each feature, (typical words or ngrams) in a document for each document class is computed in the training phase, using the Bayes Theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B} \tag{1}$$

This work was self funded, Dr Adams is employed in a Back End Developer Role not related to this work

Corresponding author:
77 Twyford Abbey Road, London NW10 7ET
Email: barry.david.adams@gmail.com

. When new documents are presented, the class of the document c, is found to be

$$c = \arg\max_c P(c). \prod_{i=1}^{m} P(x_i|c) \tag{2}$$

We product is typical turned into a sum by taking the logarithm of the probability. It can be seen that speed of the classifier depends on the product of number of features in document and the number of document classes in the training set.

History of the Algorithm

This Algorithm has first discovered by the Author while working at Magus Research, the GB Patient Application [2], was declined because Algorithms are not Patentable, the Application was published in 2004, and is thus public knowledge. We have returned the Algorithm recently, to publish the work in an Academic format and produce results on how strong the Algorithm on the Reuters 21578 data set, and a subset there of.

The Algorithm

One Central Assumption of the Naive classifier is that the probabilities of each feature in the document are independent, this is often a false assumption, if your classifying documents in the subjects of nuclear physics and ecology (say), the chance of the words 'quark' and 'proton' are highly non independent, and the naive Bayes classier will score them two strongly when both occur.

in this work we relax the (Naive) assumption that each document feature, words or ngrams, are independent, instead computing the correlation of the features, (hereafter refered to a words), in the training phase, Our SuperBayesCat classifier calculates and stores the error in the independence assumption in the training phase. If $E(Feature1, Feature)$ is 0 when the words (document features) are completely independent, and 1 when the feature always occur together.

We wish to see the error in the assumption

$$P(AB\|c) = P(A\|c)P(B\|c) \tag{3}$$

We may form $E(A, B)$

$$E(A, B) = 1 - \frac{\sum_c P(A, B\|c) - P(A|c)P(B|c)}{\sqrt{sum_c P(A, B\|c))}\sqrt{sum_c P(A\|c)P(B\|c))}} \tag{4}$$

When $E = 1$ A and B are independent and contribution $\log P(A) \log P(B)$ to the document. But when $E = 0$ A always occurs with B, in the class, and the correct contribution to the Bayes classifier sum, is just $\log P(A)$.

Thus the better measure of classification, is

$$c = \arg\max_{c} Pr(c).\sum_{A=1}^{m} 2^{-\max_{B,A\neq B}(E_{A,B})}\log(P(A|c)) \tag{5}$$

using the two fixed points, linearity and differentiability.

Where the correlation error C can be estimated in the training phase, from the probability of the features (words) A and B in the overall collection of classes, ($P(A)$ and $P(B)$, and the number of occurrences of word $Occ(A|c)$ and word $Occ(B|c)$ in the any document in class c, and the total number of words in class c, T_c) ,forming the vectors of classes, s and t,

$$t_c(A,B)) = P(A|c)P(B|c), s_c(A,B) = \frac{occ(A)occ(B)}{T_c} \tag{6}$$

$$C_{A|B} = 1 - \frac{\sum_c t_c(A,B).s_c(A,B)}{\sqrt{\sum_c()t_c(A|B))^2}\sqrt{\sum_c(s_c(A|B))^2}} \tag{7}$$

Or one minus the normalised dot product between s and t over the classes.

We can see that the speed of the classifier depends on the number of different words in the classification squared, and upon the number of different words in the test document squared. In practice the training can be slow, but the classification is still fast.

Testing

We made a fresh new write of a categorizer program which we published to github [3], We use both single words, and pairs of words one after another, doing estimation of the error in the assumption of independence $E(A,B)$ only for the single words. We introduced tunable feature detection with a minimum entropy due to the feature. Our feature detection may have an numerical bug in the program as the entropy bits for reasonable number of words, seem a little off.

The test set was the Reuter 21578 Apte collection with 90 categories and 9,598 training documents. We also used a same 10 common classes as [5] Keyvanpour et al, as a subset.

We however able to tune for around 10000 words, and 30000 word pairs for the 10 classes, by arranging the minimum entropies., we then used the same entropy settings for full 90 Reuters classes.

References

[1] Pouria Kaviani, Mrs Sunita Dhotre, *Short Survey on Naivce Bayes Algorithm*, Internation Journal of Advance Engineering and Research Development, (2017)

[2] Barry Adams Patient Application BG2395301A *A method of classifying a set of data* App GB0226145A·2002-11-08 Publication GB2395301A·2004-05-19, (2004), https://worldwide.espacenet.com/patent/search/family/009947499/publication/GB2395301A?q=pn

[3] SuperBayesCat on github, (2021) https://github.com/badams77-cpu/SuperBayesCat/tree/master

[4] Text Categorization Corpora, http://disi.unitn.it/moschitti/corpora.htm

[5] Keyvanpour, Mohammad and Bahojb Imani, Maryam, *Semi-supervised text categorization: Exploiting unlabeled data using ensemble, learning algorithms*, Intelligent Data Analysis 05-2013 page 367-385, doi 10.3233/IDA-130584