




nb: Naive Bayes Model in R

Renato Rodrigues Silva¹

¹ Federal University of Goiás

DOI: [10.21105/joss.00918](https://doi.org/10.21105/joss.00918)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted: 27 August 2018

Published: 07 September 2018

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

One of the most important tasks in statistics is to classify experimental units and or individuals and one of the simplest algorithm to make classification is the naive Bayes (NB) (???).

The algorithm assumes each instances belongs to only one of the categories, which are finites and mutually exclusives. Moreover, it is assumed all features are independents from each other, which is unrealistic in most of the times but provides the efficiency and simplicity to the algorithm. In a overview, the algorithm classifies the instances into respectively groups, using Bayes' theorem (???, (???)).

Naive Bayes classifier has been widely used in several areas of knowledge. (???) used the Naive Bayes to classify non-vocalized Arabic web documents into pre-defined categories. They found an average accuracy over all categories of 68.78 %. where the maximum accuracy found were equal to 92.8%. (???) applied the naive Bayes to implement automatic classification of the web chinese text. Experimental results showed that the classification system were efficient and accurate. In the diagnosis medical area, (???) compared the performance of naive Bayes with others fifteen famous classifier used in the literature, they concluded that naive Bayes had high performance in most of the examined medical problems.

The naive Bayes has been implemented in many programming languages. Nowadays, it is very easy to find a tutorial about how to performance naive Bayes using python (???) or R (???)

R is a programming language and software for data analysis supported R Foundation for Statistical Computing (???). One of an advantages to use R is the large number of packages available by users and scientists around the world. Up to our knowledge, there some R packages where naive Bayes is implemented. One of the options is the Bnlearn package (???), a R package for graphical models in general. Among the various functions defined in the library, they implemented bernoulli and multinomial naive Bayes, where feature and categorical variables follows multinomial or bernoulli distribution. It is possible the users run the Gaussian naive bayes, but in that case, the users must provide the graphical model by theirselves.

Gaussian, bernoulli and multinomial naive Bayes are implemented in e1071 (???) and mlr (???) also. However, none of them allows modelling the qualitative/discrete and real valued features at the same time.

Here, it is being purposed a new R package for naive Bayes models. Standards variants of naive Bayes models: Gaussian, bernoulli and multinomial naive Bayes were implemented. However, in nb package is possbile to mixed quantitative and qualitative features. It is possible because this implementation takes advantages of the fact that features are independent each other. Hence, The package uses the normal distribution for modelling real

valued features, and multinomial / bernoulli distribution for modelling discrete / qualitative features. The library contains four functions: `naive.bayes`, `post.prob`, `classify` and `confusion.matrix`.

The `naive.bayes` function estimates parameters of naive Bayes model via maximum likelihood estimation, The input of this function is a `data.frame` `x` which represents the dataset to be analyzed and `ct`, which is a string of the name of categorical variable that classifies the instances into groups. If `x` contain only real valued features the `naive.bayes` estimates the parameters of a Gaussian naive Bayes model. On the other hand, if there are only qualitative / discrete features the parameters are estimated from Multinomial naive Bayes model, otherwise, the function uses the gaussian distribution for real valued features and multinomial / bernoulli distribution for qualitative / discrete features because the features are assumed to be independent.

The `post.prob` computes the probability posterior of a new instances belong to a class given the a set of observed features. The output is a vector with posterior probability for each category. In turn, the function `classify` computes which the category has higher posterior probability.

$$\arg \max_{j \in (1, \dots, k)} p(\mathbf{z}) \prod p(x_i | \mathbf{Z} = \mathbf{z})$$

where \mathbf{Z} the random vector that represents the categorical variables and x_i the features.

Furthermore, the `nb` packages has another advantage in comparison to other packages. It is already implemented a `confusion matrix` function for measuring the predictive performance of model. The confusion matrix is defined via 5 or 10 fold cross validation. To create the folds, the function `createFolds` of `caret` package (???) is invoked. Currently, `nb` package is hosted at [github](#)

References