The Journal of Open Source Software

FuzzyClass: A family of Fuzzy and Non-Fuzzy probabilistic-based classifiers

Jodavid A. Ferreira [1*] **and Ronei M. Moraes** [1*]

1 Department of Statistics, Federal University of Paraiba, João Pessoa, Brazil * These authors contributed equally.

DOI: 10.21105/joss.05613

Software

- Review
- Repository
- Archive

Editor: Andrew Stewart

Reviewers:

- @tianyi93
- @MikeLydeamore

Submitted: 31 March 2023
Published: 25 August 2023

Summary

This paper presents a package written in the language R for classifiers based on Naive Bayes and Fuzzy Naive Bayes named `FuzzyClass`. This R package implements eight fuzzy classifiers, with option for using the classical ones too. An example in which the Fuzzy Gaussian Naive Bayes method is presented.

Statement of need

Classification is assign labels or classes for a data set (Pathak, 2014). Several methods, are also used in pattern recognition (Webb, 2003), computational intelligence (Konar, 2006) and decision making (Efraim, 2011). The difficulties encountered in classification are also considered as one of the central problems of machine learning. However, all of them have the same goal. A special type of classification in which the class label takes on two values, that is named binary. The classification models in which the target variable has more than two values is called multiclass algorithms.

Uncertainty and imprecision are sources of problems in modeling and building classifiers. The first one can be modeled from probability theory and the second one can be modeled by fuzzy set theory, which was developed by Zadeh (1965). In fuzzy set theory, elements can belong to more than one set simultaneously with a certain degree of membership, which is a value defined in the range $[0, 1]$, which determines how much the element belongs to the fuzzy set.

Zadeh assumed that imprecision can be modeled using a fuzzy membership function on probability distributions (see more Zadeh (1988)). Several classification methods have been proposed using probability theory for fuzzy events (RM. Moraes, Soares, et al., 2020; RM. Moraes & Machado, 2006, 2014). Classifiers based on probability and Zadeh's probability were implemented using the Binomial distribution (RM. Moraes & Machado, 2016a), the Poisson distribution (RM. Moraes & Machado, 2015), the Beta distribution (RM. Moraes, Rodrigues, et al., 2020), the Exponential distribution (RM. Moraes & Machado, 2016b), the Gamma distribution (RM. Moraes et al., 2018), the Gaussian distribution (RM. Moraes & Machado, 2010), the Triangular distribution (R. Moraes et al., 2020) and Trapezoidal distribution (Lopes et al., 2023). These classifiers were implemented in the `R` and made available through a package named `FuzzyClass`, which will be the basis of this article and can be found at the link: https://cran.r-project.org/web/packages/FuzzyClass/. Classifiers such as Naive Bayes, Gaussian Naive Bayes, Bernoulli Naive Bayes, and Poisson Naive Bayes can be found in libraries and software like `scikit-learn` (`python`), `Weka`, and `R` packages `naivebayes` and `e1071`. However, none of them offer implementations with fuzzy. All implementations involving fuzzy probability and distributions not mentioned earlier are contributions provided by this package. It is worth noting that these works were developed in the LabTEVE (http://www.de.ufpb.br/~labteve/) and LEAPIG (http://www.de.ufpb.br/~leapig/) research laboratories, both at Federal University of Paraiba, Brazil.

Statistical Modeling and Discrimination Measures

The classifiers presented in this paper are divided between distributions for discrete and for continuous variables.

Naive Bayes and Fuzzy Naive Bayes

In this section it is assumed that the random variables for the data are multivariate and they are represented by $\mathbf{x}$. Thus, let $\mathbf{x}_i = \{X_{i1}, X_{i2}, \ldots, X_{ik}\}$ be a random vector of data in the i-th sample with k-information (dimension/variables) obtained from training data and $w_j, j \in \Omega$ is the real class for $\mathbf{x}$. Let $\Omega = 1, ..., M$ be the total number of classes, denoted by M. The probability of the class w_j assuming that each variable X_{it} is conditionally independent of any other variable X_{il} for all $t \neq l \leq k$, is:

$$P(w_j|X_{i1}, X_{i2}, ..., X_{ik}) = \frac{1}{S} P(w_j) \prod_{t=1}^{k} P(X_{it}|w_j).$$

The Fuzzy Naive Bayes Network

The Fuzzy Naive Bayes Networks are based on the Zadeh's definition of probability of fuzzy events (Zadeh, 1968). Thus, let membership function $\mu_j(X_{it})$ for the variable X_{it}, and class w_j, the Zadeh's probability for this class is:

$$\mathcal{P}\left(w_j|X_{i1}, X_{i2}, ..., X_{ik}\right) = \frac{1}{S} P(w_j) \prod_{t=1}^{k} P(X_{it}|w_j)\mu_j(X_{it}).$$

As criterion the decision of the classifier, we have that the vector $\mathbf{x_i}$ will be assigned to the class that

$$\hat{w}_j = \arg max_{j \in \Omega} P(w_j|\mathbf{x}_i) \quad \text{and} \quad \hat{w}_j = \arg max_{j \in \Omega} \mathcal{P}(w_j|\mathbf{x}_i).$$

where $P(w_j|\mathbf{x}_i)$ will have as a probability function or pdf assuming the distributions Binomial, Beta, Exponential, Gamma, Gaussian, Poisson, Triangular, and Trapezoidal distributions.

Motivating examples

Package functions need input arguments, some of which will be described below and others can be consulted in the package's documentation. So, follow:

- *train* that is a matrix or data frame of training set cases;
- *cl* factor of true classifications of training set;
- *fuzzy* boolean variable to use or not the membership function;

In the example below, an application with real data will be presented using data from the paper (RM. Moraes & Machado, 2010), appliying the classifier Fuzzy Gaussian Naive Bayes, which in the package has the nomenclature of `FuzzyGaussianNaiveBayes`.

The data presented below were used for performance evaluation in a virtual reality (VR) simulator in that paper. Three classes of performance were defined by the expert and numbered (M=3): correct procedures (1), acceptable procedures (2) and badly executed procedures

(3). Then, the classes of performance for a trainee could be: "you are well qualified", "you need some training yet" and "you need more training". Thus, our following example has three distinct classes, as can be seen in the following variable V4:

```
R> library(FuzzyClass)
R> data(VirtualRealityData)
R> head(VirtualRealityData)
```

```
         V1     V2      V3 V4
308 13.7027 7.3439 10.9141  2
183  1.8535 8.1123  8.5844  1
591 16.3139 9.9005 14.2228  3
12   1.5508 6.0448  8.2070  1
231  6.1457 8.6309 13.4432  2
254 12.0941 9.5665 14.1032  2
```

When classifying using `FuzzyGaussianNaiveBayes()` we have:

```
R>split <- caTools::sample.split(t(VirtualRealityData[,1]),
+                                 SplitRatio = 0.75)
R> Train <- subset(VirtualRealityData, split == "TRUE")
R> Test <- subset(VirtualRealityData, split == "FALSE")
R> target <- Train[, 4]
R> features <- Train[, -4]
R> fit_FGNB <- FuzzyGaussianNaiveBayes(train = features,
+                                       cl = target,cores = 2)
R> targetTest <- as.factor(Test[,4])
R> pred_FGNB <- predict(fit_FGNB, Test[,-4])
```

```
R> result <- caret::confusionMatrix(targetTest, pred_FGNB)
R> # confusionMatrix
R> result$table
```

```
          Reference
Prediction  1  2  3
         1 50  2  0
         2  2 44  1
         3  0  7 44
```

```
R> result$overall[1]
```

```
Accuracy
    0.92
```

```
R> result$overall[2]
```

```
    Kappa
0.8800799
```

The funcion `fit_FGNB` estimates distribution parameters, membership functions. Those results can be accessed by the user using `fit_FGNB$medias`, `fit_FGNB$varian`, and `fit_FGNB$Pertinencias`, respectively.

The function `predict` contains all the predicted classes. The probabilities for each sample, can be accessible for the user, using the input parameter `type="matrix"`.

Through this example, which was also the result of published articles, steps can be followed and classifiers can be applied to other data. As well as the different classifiers following the same structure of prediction of the classes. For more detailed help for each classifier, the package manual can be found at the following link: https://cran.r-project.org/web/packages/FuzzyClass/FuzzyClass.pdf.

Acknowledgements

This project is partially supported by grant 310470/2012-9 of the National Council for Scientific and Technological Development (CNPq). Jodavid A. Ferreira has been supported by grant 1278/2021 of the Paraíba State Research Foudation (FAPESQ).

References

Efraim, T. (2011). *Decision support and business intelligence systems*. Pearson Education India. https://doi.org/10.1002/9780470634431

Konar, A. (2006). *Computational intelligence: Principles, techniques and applications*. Springer Science & Business Media. https://doi.org/10.1093/comjnl/bxm073

Lopes, ARR., Ferreira, JA., Machado, LS., & Moraes, RM. (2023). A new fuzzy trapezoidal naive bayes network as basis for assessment in training based on virtual reality. *Machine Learning, Multi Agent and Cyber Physical Systems: Proceedings of the 15th International FLINS Conference (FLINS 2022)*, 600–607. https://doi.org/10.1142/9789811269264_0071

Moraes, RM., & Machado, LS. (2016a). A fuzzy binomial naive bayes classifier for epidemiological data. *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 745–750. https://doi.org/10.1109/fuzz-ieee.2016.7737762

Moraes, RM., & Machado, LS. (2016b). A fuzzy exponential naive bayes classifier. *Uncertainty Modelling in Knowledge Engineering and Decision Making: Proceedings of the 12th International FLINS Conference*, 207–212. https://doi.org/10.1142/9789813146976_0035

Moraes, RM., & Machado, LS. (2015). A fuzzy poisson naive bayes classifier for epidemiological purposes. *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, *2*, 193–198. https://doi.org/10.5220/0005642801930198

Moraes, RM., & Machado, LS. (2006). On-line training evaluation in virtual reality simulators using fuzzy bayes rule. In *Applied artificial intelligence* (pp. 791–798). World Scientific. https://doi.org/10.1142/9789812774118_0111

Moraes, RM., & Machado, LS. (2010). Fuzzy gaussian naive bayes applied to online assessment in virtual reality simulators. In *Computational intelligence: Foundations and applications* (pp. 243–248). World Scientific. https://doi.org/10.1142/9789814324700_0035

Moraes, RM., & Machado, LS. (2014). Psychomotor skills assessment in medical training based on virtual reality using a weighted possibilistic approach. *Knowledge-Based Systems*, *70*, 97–102. https://doi.org/10.1016/j.knosys.2014.05.006

Moraes, RM., Rodrigues, AKG., Soares, EAMG., & Machado, LS. (2020). A new fuzzy beta naive bayes classifier. *Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020)*, 437–445. https://doi.org/10.1142/9789811223334_0053

Moraes, RM., Soares, EAMG., & Machado, LS. (2018). A fuzzy gamma naive bayes classifier. *Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018)*, 691–699. https://doi.org/10.1142/9789813273238_0088

Moraes, RM., Soares, EAMG., & Machado, LS. (2020). A double weighted fuzzy gamma naive bayes classifier. *Journal of Intelligent & Fuzzy Systems*, *Preprint*, 1–12. https://doi.org/10.3233/jifs-179431

Moraes, R., Silva, ILA., & Machado, LS. (2020). Online skills assessment in training based on virtual reality using a novel fuzzy triangular naive bayes network. *Proc. FLINS*, 446–454. https://doi.org/10.1142/9789811223334_0054

Pathak, MA. (2014). *Beginning data science with R*. Springer. https://doi.org/10.1007/978-3-319-12066-9

Webb, AR. (2003). *Statistical pattern recognition*. John Wiley & Sons. https://doi.org/10.1002/0470854774

Zadeh, LA. (1965). Information and control. Fuzzy sets. *Information and Control*, *8*(3), 338. https://doi.org/10.1016/s0019-9958(65)90241-x

Zadeh, LA. (1968). Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, *23*(2), 421–427. https://doi.org/10.1016/0022-247x(68)90078-4

Zadeh, LA. (1988). Fuzzy logic. *Computer*, *21*(4), 83–93. https://doi.org/10.1109/2.53