Building an Ensemble of Naive Bayes Classifiers using committee of bootstraps and monte carlo splits for a various percentage of random objects from training set

Piotr Artiemjew, Pawel Idzikowski

Faculty of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn
Poland
email:artem@matman.uwm.edu.pl, iddziku@gmail.com

Abstract. In the work we have implemented an ensemble of Naive Bayes classifiers using committee of bootstraps and monte carlo splits. We have conducted 50 iterations of learning in each tested model. Fixed percentage of random objects from the original training system was used. New training decision systems that were considered consisted of 10 to 100 percent of random objects from original training decision system. Two main variants were checked, first with objects returning after the drawn (bootstraps) - and without returning (as monte carlo splits). We have presented how Naive Bayes classifier works in mentioned models on selected data from UCI repository.

Keywords: Ensemble model, Naive Bayes Classifier, Bootstrap, Monte carlo split,Decision Systems, Classification

1 Introduction

The ensemble scheme of classification is really effective in many contexts, for instance in rough set methods the exemplary successful applications can be found in [1, 2, 8, 9, 12, 17, 20]. In the work we are trying to answer the question of how the fixed percentage of drawn objects from the original training set can influence the ensemble of Naive Bayes (NB) classifiers. We have implemented two variants for committees - bootstrap and monte carlo split. In Sect. 2 we have introduced theory and show toy examples for Naive Bayes classifier. In Sect 3 we have brief introduction to used Ensemble models. In Sect. 4 we show the experiment settings and in Sect. 5 the results of the experiments. We conclude the paper in Sect. 6. Let us to start with basic knowledge about used classifier [15].

2 Naive Bayes classifier

The Bayes classifier, for a general perspective, cf., Mitchell [11], Devroye et al.[6], Duda et al. [7], or Bishop [4], for monographic expositions, and Langley et al. [10], and, Rish et al. [16] for analysis of classifier performance vs. data structure, was introduced in Ch. 3. Its study in rough set framework was, given, e.g., in Pawlak [14], Al–Aidaroos et al. [3], Cheng et al. [5], Su et al. [21], Wang et al. [23], [24], Yao and Zhou [26], Zhang et al. [27].

Naive Bayes classifier owes its naivety epithet to the fact that one assumes the independence of attributes which condition in reality is not often met. Its working in the realm of decision systems can be described concisely as follows. For a given training decision system (U_{trn}, A, d) and a test system (U_{tst}, A, d), where $U = U_{trn} \cup U_{tst}$ is the set of objects, $A = \{a_1, a_2, ..., a_n\}$ is the conditional attribute set, and d is the decision attribute.

The classification of a test object $v \in U_{tst}$ described by means of its information set $(a_1(v), a_2(v), ..., a_n(v))$ consists of computing for all decision classes the value of the parameter

$$P(d = d_i | b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v))$$

and the decision on v is the decision value with the maximal value of the parameter.

The Bayes theorem along with the frequency interpretation of probability allows to express this probability as

$$\frac{P(b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v) | d = d_i) * P(d = d_i)}{P(b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v))} \tag{1}$$

One usually dispenses with the denominator of equation 1, because it is constant for all decision classes. Assuming independence of attributes, the numerator of equation 1 can be computed as

$$P(d = d_i) * \prod_{m=1}^{n} P(b_m = a_m(v) | d = d_i).$$

In practice, we can use partial estimation

$$P(b_m = a_m(v) | d = d_i) = \frac{\text{number of test instances } b_m = a_m(v) \text{ in training class } d_i}{\text{cardinality of class } d_i}.$$

Each decision class is voting by submitting the value of the parameter

$$Param_{d=d_i} = P(d = d_i) * \prod_{m=1}^{n} P(b_m = a_m(v) | d = d_i). \tag{2}$$

In this approach, we could encounter a problem of zero frequency of a descriptor $b_m = a_m(v)$ in a class d_i, i.e., $P(b_m = a_m(v) | d = d_i) = 0$. One of the methods to avoid the problem of zero–valued decisive parameters, is to search among the remaining classes for the smallest non–zero value of $P(b_m = a_m(v)|d = d_j)$. The found value is additionally slightly lowered, and assigned instead of the zero value. In case of more than one class with the zero frequency of the descriptor $b_m = a_m(v)$, we could assign such reduced value to all of them. Another method to avoid this problem is to consider the remaining decision classes, which contain the value $b_m = a_m(v)$. In case of the zero frequency of the descriptor $b_m = a_m(v)$ in all training classes, this descriptor can be disregarded. In order to help ourselves with the task of computing with small numbers, we can use logarithms of probabilities. In practice, it is also acceptable to use sums instead of products in which case decision classes vote by the parameter

$$Param_{d=d_i} = P(d = d_i) * \sum_{m=1}^{n} P(b_m = a_m(v)|d = d_i). \tag{3}$$

This classifier is fit for symbolic attributes. In case of numerical data, assuming the normal distribution, the probability $P(b_m = a_m(v)|d = d_i)$ can be estimated based on the Gaussian function

$$f(x) = \frac{1}{\sqrt{(2 * \pi * \sigma_c^2)}} * e^{\frac{-(x-\mu_c)^2}{2*\sigma_c^2}}.$$

To compute this value, the estimates of mean values and variances in decision classes are necessary:

$$\mu_c = \frac{\sum_{i=1}^{cardinality\ of\ class\ c} a(v_i)}{cardinality\ of\ class\ c},$$

$$\sigma_c^2 = \frac{1}{cardinality\ of\ class\ c} * \sum_{i=1}^{cardinality\ of\ class\ c} (a(v_i) - \mu_c)^2.$$

2.1 An example of Bayes classification

In this section we show an exemplary classification on the lines of equation (2). The test decision system is given as

Table 1. Test system (X, A, c)

	a_1	a_2	a_3	a_4	c
x_1	2	4	2	1	4
x_2	1	2	1	1	2
x_3	9	7	10	7	4
x_4	4	4	10	10	2

and the training system is

Table 2. Training system (Y, A, c)

	a_1	a_2	a_3	a_4	c
y_1	1	3	1	1	2
y_2	10	3	2	1	2
y_3	2	3	1	1	2
y_4	10	9	7	1	4
y_5	3	5	2	2	4
y_6	2	3	1	1	4

We have $P(c = 2) = \frac{3}{6} = \frac{1}{2}$, $P(c = 4) = \frac{1}{2}$.
We start with classification of the test object x_1 whose information set is $(2, 4, 2, 1)$ and the decision $c = 4$.

According to the formula in (**??**), we obtain
$P(a_1 = 2|c = 2) = \frac{1}{3}$.
$P(a_2 = 4|c = 2) = \frac{0}{3}$ we cannot handle it, there is no descriptor $a_2 = 4$ in all classes. Next,

$P(a_3 = 2|c = 2) = \frac{1}{3}$.
$P(a_4 = 1|c = 2) = \frac{3}{3}$.

Finally, $Param_{c=2} = \frac{1}{2} * (\frac{1}{3} + \frac{0}{3} + \frac{1}{3} + \frac{3}{3}) = \frac{5}{6}$.
Continuing, we obtain
$P(a_1 = 2|c = 4) = \frac{1}{3}$.
$P(a_2 = 4|c = 4) = \frac{0}{3}$ we cannot handle it, there is no descriptor $a_2 = 4$ in all classes.
$P(a_3 = 2|c = 4) = \frac{1}{3}$.
$P(a_4 = 1|c = 4) = \frac{2}{3}$.
Finally, $Param_{c=4} = \frac{1}{2} * (\frac{1}{3} + \frac{0}{3} + \frac{1}{3} + \frac{2}{3}) = \frac{2}{3}$.
As $Param_{c=2} > Param_{c=4}$, the object x_1 is assigned the decision value of 2. This decision is inconsistent with the expert decision, this object is incorrectly classified.

For the second test object x_2, with the information set $(1, 2, 1, 1)$ and the decision value of 4, we obtain in the analogous manner:

$P(a_1 = 1|c = 2) = \frac{1}{3}$, we increase counter by 1 because $P(a_1|c = 4) = 0$ so finally $P(a_1 = 1|c = 2) = \frac{2}{3}$.
$P(a_2 = 2|c = 2) = \frac{0}{3}$ we cannot handle it because the descriptor $a_2 = 2$ is missing in all classes.
$P(a_3 = 1|c = 2) = \frac{1}{3}$.
$P(a_4 = 1|c = 2) = \frac{3}{3}$,
so $Param_{c=2} = \frac{1}{2} * (\frac{2}{3} + \frac{0}{3} + \frac{1}{3} + \frac{3}{3}) = 1$.

$P(a_1 = 1|c = 4) = \frac{0}{3}$, in this case we have to increase counter of $P(a_1 = 1|c = 2)$ by one to account for the class, which contains at least one count of the descriptor $a_1 = 1$.

$P(a_2 = 2|c = 4) = \frac{0}{3}$, we cannot handle it, $a_2 = 2$ is missing in all classes.
$P(a_3 = 1|c = 4) = \frac{1}{3}$.
$P(a_4 = 1|c = 4) = \frac{2}{3}$, so, finally, $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{1}{3} + \frac{2}{3}) = \frac{1}{2}$.
As $Param_{c=2} > Param_{c=4}$, the object x_2 is assigned the decision value of 2; this decision is consistent with the expert decision so the object is correctly classified.

The next test object is x_3 with the information set $(9, 7, 10, 7, 4)$.
We have $Param_{c=2} = P(c = 2) * \sum_{i=1}^{4} P(a_i = v_i|c = 2)$, and,

$P(a_1 = 9|c = 2) = \frac{0}{3}$.
$P(a_2 = 7|c = 2) = \frac{0}{3}$.
$P(a_3 = 10|c = 2) = \frac{0}{3}$.
$P(a_4 = 7|c = 2) = \frac{0}{3}$,

so, finally, $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$.
Also, for $Param_{c=4} = P(c = 4) * \sum_{i=1}^{4} P(a_i = v_i|c = 4)$, we have
$P(a_1 = 9|c = 4) = \frac{0}{3}$.
$P(a_2 = 7|c = 4) = \frac{0}{3}$.
$P(a_3 = 10|c = 4) = \frac{0}{3}$.
$P(a_4 = 7|c = 4) = \frac{0}{3}$,

and, finally, $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$.
As $Param_{c=2} == Param_{c=4}$, the random decision $random(2, 4) = 4$ is assigned to x_3, so the object is correctly classified.

For the last test object, x_4 with the information set $(4, 4, 10, 10, 4)$, we compute

$Param_{c=2} = P(c = 2) * \sum_{i=1}^{4} P(a_i = v_i|c = 2)$:
$P(a_1 = 4|c = 2) = \frac{0}{3}$.
$P(a_2 = 4|c = 2) = \frac{0}{3}$.
$P(a_3 = 10|c = 2) = \frac{0}{3}$.
$P(a_4 = 10|c = 2) = \frac{0}{3}$,

and, finally, $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$.

For $Param_{c=4} = P(c = 4) * \sum_{i=1}^{4} P(a_i = v_i|c = 4)$,we need:
$P(a_1 = 4|c = 4) = \frac{0}{3}$.
$P(a_2 = 4|c = 4) = \frac{0}{3}$.
$P(a_3 = 10|c = 4) = \frac{0}{3}$.
$P(a_4 = 10|c = 4) = \frac{0}{3}$,

hence, $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$.

A random decision assignment $random(2, 4) = 4$ causes x_4 to be incorrectly classified.
We now compute parameters:

$$Global\_Accuracy = \frac{\text{number of tst objects correctly classified in whole test system}}{\text{number of classified objects in whole testsystem}};$$

$$Balanced\_Accuracy = \frac{\sum_{i=1}^{number\ of\ classes} \frac{\text{number of test objects correctly classified in class } c_i}{\text{number of objects classified in class } c_i}}{number\ of\ classes}.$$

In our exemplary case, these values are

$$Global\_Accuracy = \frac{2}{4} = \frac{1}{2};$$

$$Balanced\_Accuracy = \frac{\frac{1}{2} + \frac{1}{2}}{2} = \frac{1}{2}.$$

Tst obj	Expert decision	Decision of our classifier
x_1	4	2
x_2	2	2
x_3	4	4
x_4	2	4

3 Selected ensemble models

There are many techniques in the family of Ensemble models. One of the most popular are Random Forests, Bagging and Boosting - see [25]. Short description of used models is to be found below.

Bootstrap Ensembles - Pure Bagging: It is the random committee of bootstraps [28]. It is a method in which the original decision system - the basic knowledge - is split into (TRN) training data set, and $(TSTvalid)$ validation test data set. And from the TRN system, for a fixed number of iterations, we form a new Training systems $(NewTRN)$ by random choice with returning of $card\{TRN\}$ objects. In all iterations we classify the TRNvalid system in two ways: the first based on the actual $NewTRN$ system and the second based on the committee of all performed classifications. In the committee majority voting is performed and the ties are resolved randomly.

Committee of Monte Carlo splits: Classification method used in this algorithm is similar to the previously described with the difference that the NewTST are formed in a different way - see [13], [19] and [29]. Objects for NewTRN are simply random chosen without returning.

4 Experimental Session settings

In the next subsections we present information on how the models described above are used in our experiments.

4.1 Committee of Monte Carlo splits for NB Classifier

We have carried out a series of experiments using Australian credit data set form University of Irvine repository [22]. The original decision-making system was split by 20 to 80 percent for tst and trn sets respectively. In the case of iterations the monte carlo model, when creating new training systems, objects are randomly drawn without return. Each draw is followed by a classification of tst by a single classifier and a committee of previously learned classifiers. We ran 10 tests, where for Test i: we consider $i * 10$ percent of random objects.

4.2 Committee of Bootstraps with NB classifier

The method [28] works in the same way as described above, but the only difference is that the objects are returned after the draw and it is possible to see the copies in the training systems. The other experimental settings are identical.

As the base classifier we used the NB classifier from Sect. 2 for symbolic data. Effectiveness is assessed by the accuracy of the classification - expressed as a percentage of correctly classified objects.

5 Results of experiments

In Figs. 1 to 10 we have the classification results based on the training systems formed from 10 to 100 percent random objects of the original training system. There are two variants presented - on the left-hand side a draw without a return - on the right-hand side a draw with a return. Additionally, in Tables 3 and 4 we present the average result of 50 iterations of learning with additional parameters of assessment of the quality of classification. We used our own implementations to carry out the tests.

5.1 Discussion of results for NB

From the results we can conclude that single classifiers may work unstable when they are based on a small part of the original training system. That is, when sets of objects from individual iterations do not overlap. Another reason for their instability of single classifications may be the appearance of copies of objects for appropriately larger training systems in the return variant. The classification committee, starting from 20 iterations, starts to work steadily even for the only 10 percent of the drawn objects. The classification committee seems to be slightly more stable for the monte carlo split but its in the range of standard deviation of results. Individual classifications from individual iterations with the help of larger training systems are much more stable in the case of the monte carlo technique. But this does not have a major impact on the final effectiveness of the classification committee.

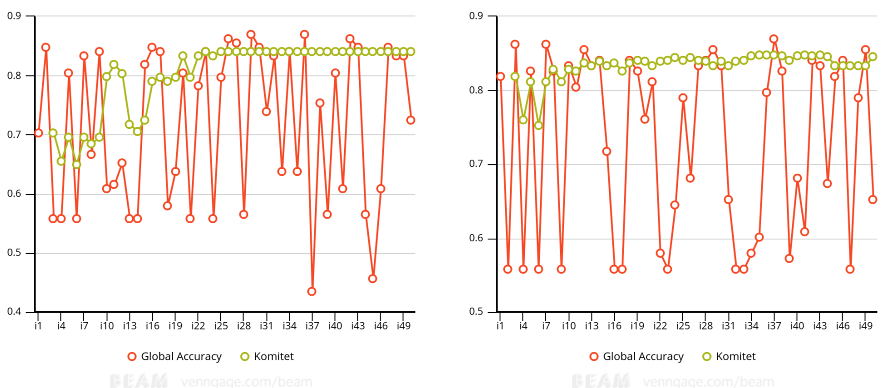

Fig. 1. tst20% - trn10%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

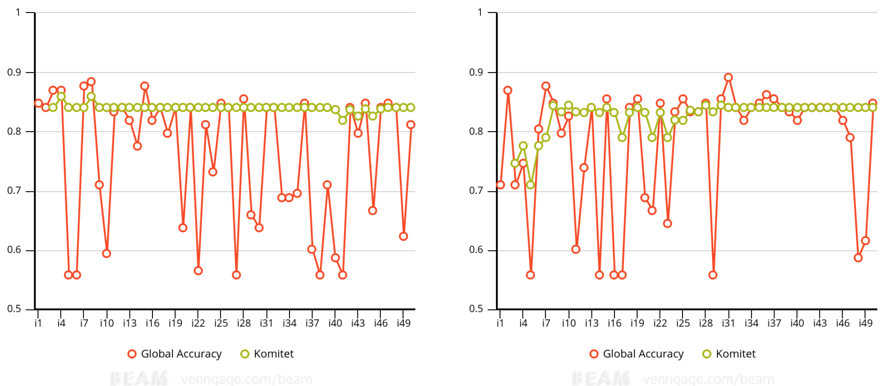

Fig. 2. tst20% - trn20%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

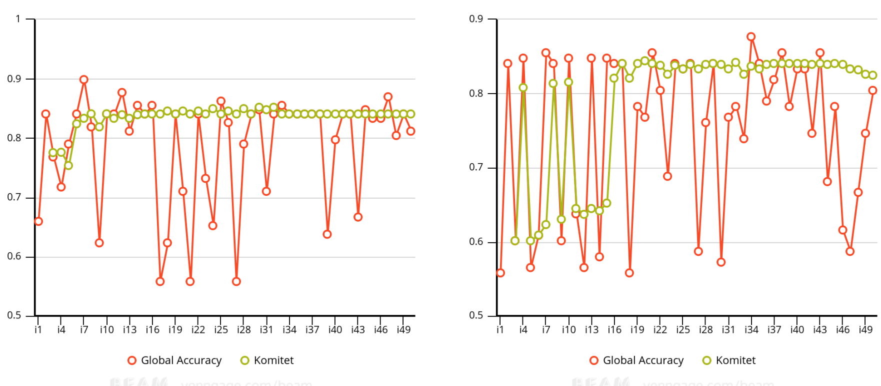

Fig. 3. tst20% - trn30%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

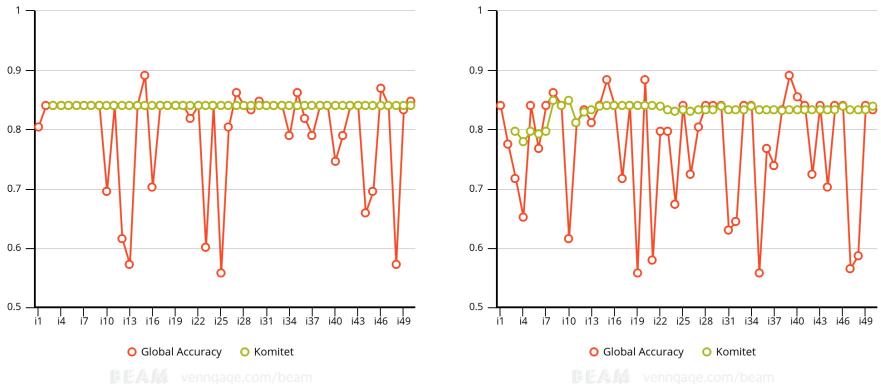

Fig. 4. tst20% - trn40%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

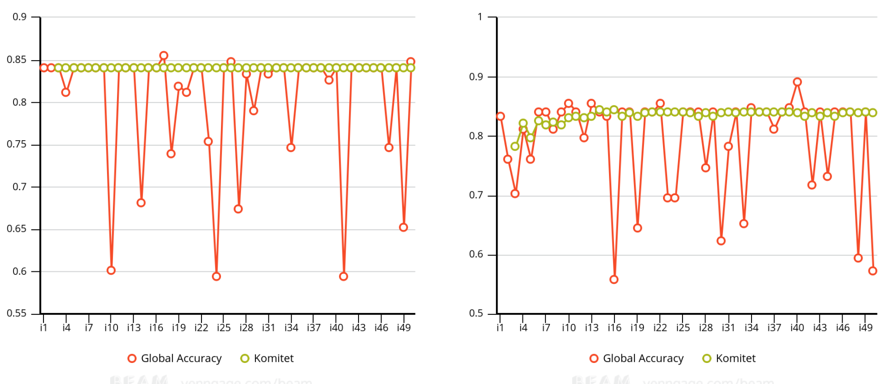

Fig. 5. tst20% - trn50%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

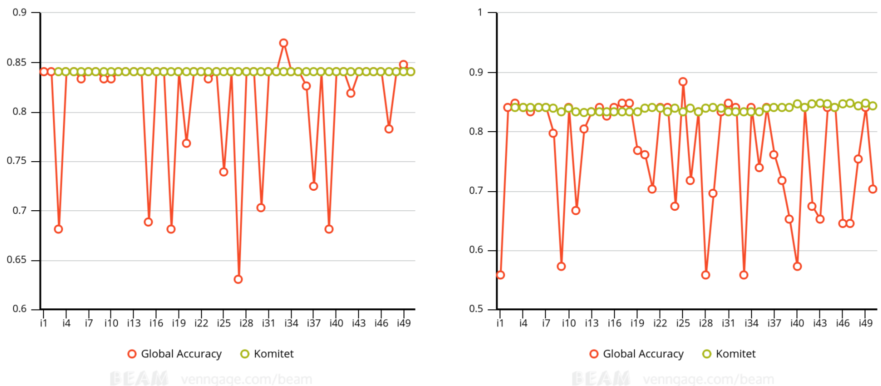

Fig. 6. tst20% - trn60%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

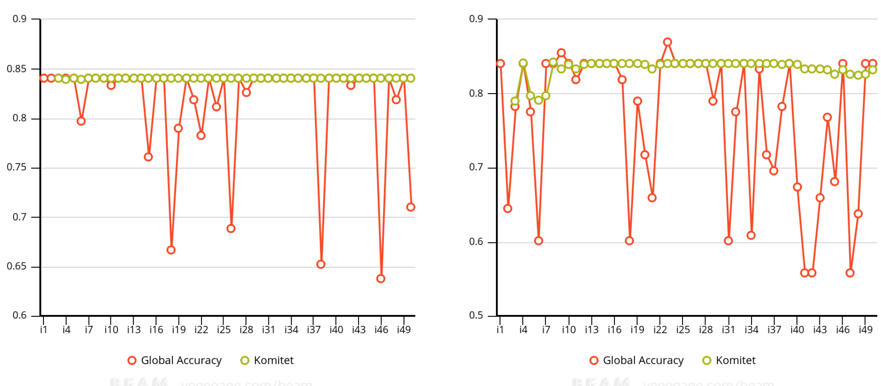

Fig. 7. tst20% - trn70%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

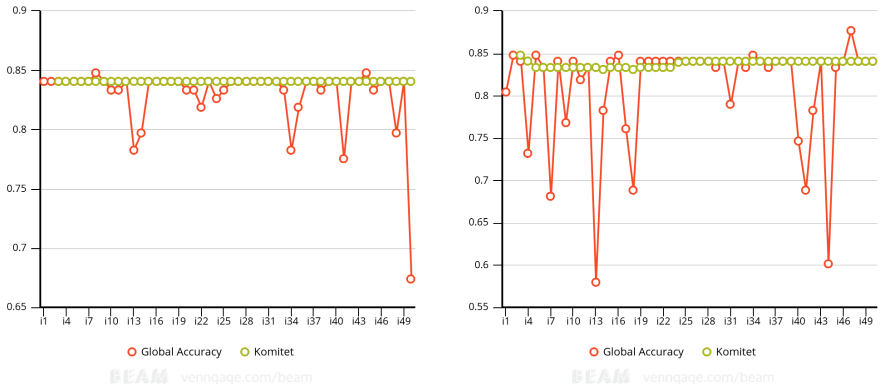

Fig. 8. tst20% - trn80%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

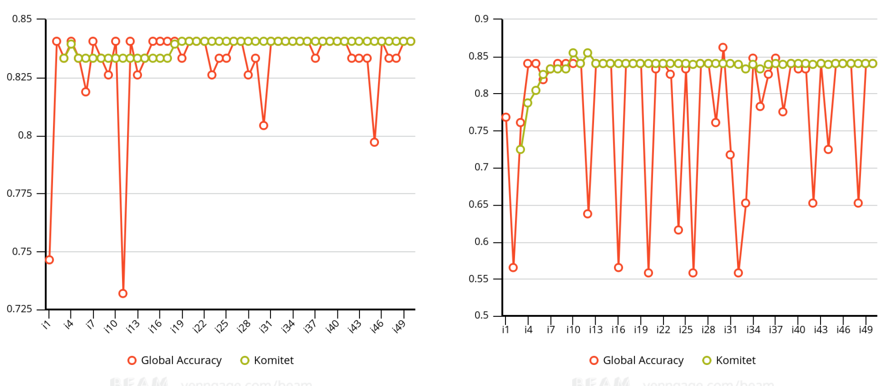

Fig. 9. tst20% - trn90%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

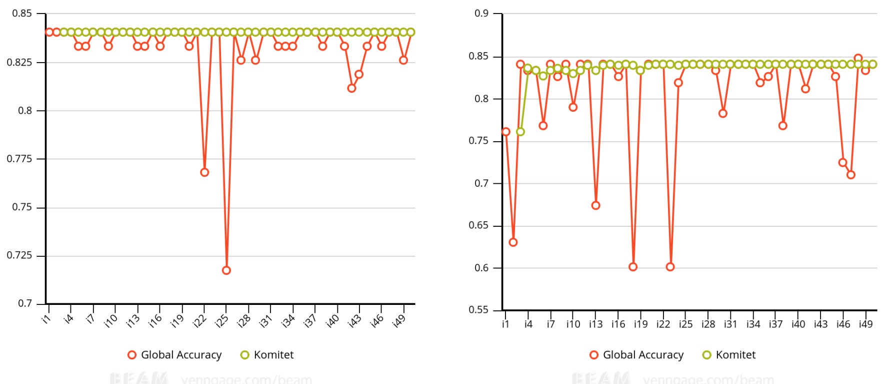

Fig. 10. tst20% - trn100%: no returning vs. with returning: result for australian credit dataset - the accuracy of classification - 50 iterations of learning

Table 3. An average effectiveness from 50 iterations, in each test we have 20 percent of tst, for Test i, trn=i*10% random objects without returning; *Global_Accuracy* is the percentage of correctly classified objects, *Global_Coverage* is percentage of classified objects, TPR_0 and TPR_1 is the precision of the classification in class 0,1 respectively,

	$Test1$	$Test2$	$Test3$	$Test4$	$Test5$	$Test6$	$Test7$	$Test8$	$Test9$	$Test10$
$Global\_Accuracy$	0.71	0.76	0.79	0.80	0.81	0.81	0.82	0.83	0.83	0.83
$Global\_Coverage$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TPR_0	0.73	0.73	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.79
TPR_1	0.72	0.85	0.88	0.93	0.95	0.95	0.95	0.95	0.94	0.94

6 Conclusions

In this experimental work we checked the performance of the Naïve Bayes classifier in the context of the classification committees based on a fixed percentage of objects drawn from the training system. We used two techniques to create training systems in particular iterations - the first one is based on the monte carlo split, where objects are drawn without returning them - the second one is based on the bootstrap model, where objects are returned. It turned out that

Table 4. An average effectiveness from 50 iterations, in each test we have 20 percent of tst, for Test i, trn=i*10% random objects with returning

	$Test1$	$Test2$	$Test3$	$Test4$	$Test5$	$Test6$	$Test7$	$Test8$	$Test9$	$Test10$
$Global\_Accuracy$	0.73	0.78	0.75	0.77	0.79	0.76	0.77	0.81	0.78	0.81
$Global\_Coverage$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TPR_0	0.73	0.75	0.73	0.74	0.75	0.72	0.73	0.77	0.74	0.76
TPR_1	0.71	0.83	0.89	0.91	0.94	0.91	0.90	0.94	0.89	0.95
$YoudenIndex$	0.44	0.59	0.62	0.65	0.68	0.63	0.62	0.71	0.63	0.71

the stability of individual classifiers increases in the case of the monte carlo method (compared to the bootstrap method) with the increase in the size of random training systems. In the case of bootstraps, increasing the training system causes more and more copies of objects, which apparently disturbs the NB classification. We observed that for the examined system, up to 10 percent of random objects used in the committee are finally starting to work steadily and give classification results comparable to those of the whole original training system. In each of the tests the classification committee starting from about the twentieth iteration begins to give good, stable results.

In future works, we plan to check the detected irregularities in other decision-making systems. We will try to check the Bayes classification under similar conditions in the context of other ensemble methods. And we plan to test the behavior of other selected classifiers.

7 Acknowledgements

The research has been supported by grant 23.610.007-300 from Ministry of Science and Higher Education of the Republic of Poland.

References

1. Artiemjew, P.: Boosting Effect of Classifier Based on Simple Granules of Knowledge, In: Information technolojy and control, Print ISSN: 1392-124X, Vol 47, No 2 (2018)
2. P., Artiemjew, K., Ropiak, “A Novel Ensemble Model - The Random Granular Reflections,” In: Proceedings of 27th international Workshop on Concurrency, Specification and Programming, CS&P 2018, Humboldt University report and published at CEUR 2018
3. Al–Aidaroos, K., Abu Bakar, A. and Othman, Z.: Data classification using rough sets and naive Bayes. In Proceedings of Int. Conference on Rough Sets and Knowledge Technology RSKT 2010, Lecture Notes in Computer Science vol. 6401, pp 134–142 (2010)
4. Bishop, Ch.: Pattern Recognition and Machine Learning, Springer-Verlag (2006)
5. Cheng, K., Luo, J. and Zhang, C.: Rough set weighted naive Bayesian classifier in intrusion prevention system. In Proceedings of the Int. Conference on Network Security, Wireless Communication and Trusted Computing NSWCTC 2009, Wuhan, P. R. China, 2009, IEEE Press, pp 25–28 (2009)

6. Devroye, L., Gy orfi, L. and Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer Verlag, New York (1996)
7. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley & Sons, New York (1973)
8. Hu, X., Construction of An Ensemble of Classifiers based on Rough Sets Theory and Database Operations, Proc. of the IEEE International Conference on Data Mining (ICDM2001), (2001)
9. Hu, X.: Ensembles of classifiers based on rough sets theory and set-oriented database operations, Presented at the 2006 IEEE International Conference on Granular Computing, Atlanta, GA (2006)
10. Langley, P., Iba, W. and Thompson, K.: An analysis of Bayesian classifiers. In Proceedings of the 10th National Conference on Artificial Intelligence, San Jose CA, 1992. AAAI Press, pp 399–406 (1992)
11. Mitchell, T.: Machine Learning. McGraw–Hill, Englewood Cliffs (1997)
12. Murthy, C., Saha, S., Pal, S.K.: Rough Set Based Ensemble Classifier, In: Rough Sets, Fuzzy Sets, Data Mining and Granular Computing Lecture Notes in Computer Science Volume 6743, p. 27 (2001)
13. Ohno-Machado, L.: Cross-validation and Bootstrap Ensembles, Bagging, Boosting, Harvard-MIT Division of Health Sciences and Technology, http://ocw.mit.edu/courses/health-sciences-and-technology/hst-951j-medical-decision-support-fall-2005/lecture-notes/hst951_6.pdf HST.951J: Medical Decision Support, Fall (2005)
14. Pawlak, Z.: Bayes theorem: The rough set perspective. In Inuiguchi, M., Tsumoto, S., Hirano, S.(eds.) : Rough Set Theory and Granular Computing, Springer Verlag, Heidelberg, pp 1–12 (2003)
15. Polkowski, L., Artiemjew, P.: Granular Computing in Decision Approximation - An Application of Rough Mereology, in: Intelligent Systems Reference Library 77, Springer, ISBN 978-3-319-12879-5, 1-422 (2015)
16. Rish, I., Hellerstein, J., Thathachar, J.: An analysis of data characteristics that affect naive Bayes performance. IBM Tech. Report RC 21993 (2001)
17. Saha, S., Murthy, C.A., Pal, S.K.: Rough set based ensemble classifier for web page classification. Fundamenta Informaticae 76(1-2), 171–187 (2007)
18. Schapire, R.E.: A Short Introduction to Boosting (1999)
19. Schapire, R.E.: The Boosting Approach to Machine Learning: An Overview, MSRI (Mathematical Sciences Research Institute) Workshop on Nonlinear Estimation and Classification (2003)
20. Shi, L., Weng, M., Ma, X., Xi, L.: Rough Set Based Decision Tree Ensemble Algorithm for Text Classification, In: Journal of Computational Information Systems6:1, 89-95 (2010)
21. Su, H., Zhang, Y., Zhao, F. and Li, Q.: An ensemble deterministic model based on rough set and fuzzy set and Bayesian optimal classifier. International Journal of Innovative Computing, Information and Control 3(4), pp 977–986 (2007)
22. University of California, Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php
23. Wang, Y., Wu, Z. and Wu, R.: Spam fitering system based on rough set and Bayesian classifier. In Proceedings of Int. IEEE Conference on Granular Computing GrC 2008, Hangzkou, P. R. China, pp 624–627 (2008)
24. Wang, Z., Webb, G. I. and Zheng, F.: Selective augmented Bayesian network classifiers based on rough set theory. In Int. Conference on Advances in Knowledge Discovery and data Mining, Lecture Notes in Computer Science vol. 3056, pp 319–328 (2004)

25. Yang, P., Yang, Y., H., Zhou, B., B.; Zomaya, A., Y.: A review of ensemble methods in bioinformatics: Including stability of feature selection and ensemble feature selection methods. In Current Bioinformatics, 5, (4):296-308, 2010 (updated on 28 Sep. 2016)
26. Yao, Yiyu, Zhou, B.: Naive Bayesian rough sets. In Proceedings of the Int. Conference on Rough Sets and Knowledge Technology RSKT 2010, Lecture Notes in Computer Science vol. 6401, pp 719–726 (2010)
27. Zhang, H., Zhou, J., Miao, D. and Gao, C.: Bayesian rough set model : A further investigation. International Journal of Approximate Reasoning 53, pp 541–557 (2012)
28. Zhou, Z.-H.: Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC. p. 23. ISBN 978-1439830031. The term boosting refers to a family of algorithms that are able to convert weak learners to strong learners (2012)
29. Zhou, Z.-H.: Boosting 25 years, CCL 2014 Keynote (2014)