Active Hidden Naive Bayes

Vangjel T. kazllarof*
Department of Mathematics, University of Patras, 26504
Rio Achaia, Greece

Sotiris B. Kotsiantis
Department of Mathematics, University of Patras, 26504
Rio Achaia, Greece

ABSTRACT

Over the years, many learners that take advantage of the Bayesian theory have been developed and proved to be both efficient and performant in terms of classification predictiveness. Hidden Naive Bayes is no exception since its polynomial complexity makes it a desired base classifier to conduct under Weakly Supervised Learning that, unlikely the Supervised Learning, takes advantage of both Labeled and Unlabeled instances in order to create accurate learning models. In this work, we exploit Hidden Naive Bayes under Active Learning scheme, where human interaction is needed for resolving the more disambiguous cases and integrating its knowledge into the learning loop. We compare the proposed Active Learner against 4 state-of-the-art classifiers under the same learning strategy over 14 binary and multiclass datasets.

CCS CONCEPTS

• **Computing methodologies** → Machine learning.

KEYWORDS

Hidden Naive Bayes, Active Learning

ACM Reference Format:
Vangjel T. kazllarof and Sotiris B. Kotsiantis. 2020. Active Hidden Naive Bayes. In *24th Pan-Hellenic Conference on Informatics (PCI 2020), November 20–22, 2020, Athens, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3437120.3437270

1 INTRODUCTION

Bayesian theory consists one of the most explored fields in Machine Learning (ML). A whole family of classifiers have been developed using this theory due to its simplicity, performance in terms of speed and their ability to solve quite successfully several classification problems, like patient age classification using electro-cardiograms [1], performing admirably to the one that their features satisfy the conditional independence property. Bayesian network classifiers [2] is the most representative member of this family and consists a structural network that can be represented as a directed graph whose vertices and edges describe the attribute and their underlying dependencies, respectively. These dependencies are measured though conditional probabilities with the given parent for each vertex. Bayesian network classifiers can be described by Equation 1, where c is the class of each instance $x = \{x_1\ x_2\ x_3 \dots\ x_k\}$ and x_i is the value of the corresponding attribute.

Assuming now that all attributes are independent, the resulted classifier will be the well-known Naive Bayes (NB), the simplest Bayesian network. However, in real world the attributes are hardly ever independent creating the need to create classifiers that takes under consideration this limitation. Tree Augmented Naive Bayes (TAN)

$$c(x) = arg \max_{c \in C} P(c) * P(x_1\ x_2\ x_3 \dots\ x_k | c) \quad (1)$$

is an improvement of NB that does not assume the attributes independency, but it permits each attribute node to have only one other attribute node as a parent, instead. However, it lacks of efficiency since it is proven that creating an optimal Augmented Naive Bayes (ABN), in which TAN is member, is an NP-hard problem since it is equivalent to creating an optimal Bayesian network [3].

Hidden Naive Bayes (HNB) is proven to be an improvement by ignoring the attribute dependencies. It uses a different structure from the default Bayesian networks introducing a hidden parent for each attribute that it is connected with and it capsules information about the influence of the other attributes [4]. This way it grows in efficiency and can be used in large scale problems where a fast and performant learner is a prerequisite. Moreover, it makes it a great candidate to Weakly Supervised Learning (WSL) tasks like Semi-Supervised Learning where real-life problems can be tackled under the existence of restricted available labeled instances, while the number of the unlabeled may be plentiful [5]. In this work HNB will be exploited under Active Learning (AL) scheme using 14 datasets from UCI repository and it will be compared with 4 state-of-the-art classifiers under the same learning conditions. In the next section, basic concepts of the AL theory will be presented while in Section 3 our proposed algorithm will be demonstrated. The experiments and the results will be described in Section 4 and Section 5 will be referred to the conclusions and future work.

2 ACTIVE LEARNING THEORY

Between Supervised and Unsupervised Learning, where the earlier has knowledge of the outcome of the instances and the later does not, a new field emerges in which there is a big portion of instances that have unknown outcome and they are used in order to enhance the predictiveness of models created by the knowledge acquired by the few instances with known outcome. This field is called Weakly Supervised Learning or Partially Supervised Learning where its main goal is to create performant models by acquiring knowledge in a small number of instances that does not already have [6].

Active Learning (AL) is a representative subfield of them, in which the human factor is taken under consideration in knowledge acquisition of instances. Separating instances that we already know its label to Labeled (L) set and the rest to Unlabeled set (U), the

*Corresponding author: Tel.: +30 69 9705 9703; E-mail: vkazlarof@upatras.gr

PCI 2020, November 20–22, 2020, Athens, Greece

ACM ISBN 978-1-4503-8897-9/20/11…$15.00
https://doi.org/10.1145/3437120.3437270

main goal of AL is to find the most informative instances in U in order to minimize the human factor involvement in the overall process, as this labeling process might be very expensive in terms of equipment or expertize in real world problems [7].

The most known query scenarios for labeling the unlabeled instances are the following: the Membership Query Synthesis in which the learner request a "label membership" in any of the unlabeled instances, the Stream - Based Selective Sampling in which the instances of the unlabeled set are queried one by one in a stream and the learner decides whether will be labeled or discarded and the Pool-Based Sampling where all unlabeled instances in the U subset-pool are ranked by the learner in order to find the most informative and acquire their label [7]. Between those scenarios, Pool-Based Scenario is the one that is used more in real life problems [8], [9], [10].

In order to rank the instances in the Pool-Based Scenario there are a number of strategies that can be followed. One of the most used strategies is Uncertainty Sampling (UncS) in which the learner selects the instances that are the least confident to ask for labeling. This approach is successful because the least confident instances are the one that are near the decision boundary between two or more classes, separating classes in more distinguish way. Some additional strategies proposed in the literature are Query-By-Committee, Expected Model Change and Variance Reduction [11].

Over the years a number of works have been published exploiting a number of classifiers under AL scheme. Support Vector Machines using AL scheme have been proposed in [12] for multi-label classification and in [13], [14] for image classification. In [15], Rotation Forest, a well-known ensemble classifier, have been proven to be a promising learner using AL compared to other ensemble classifiers.

3 PROPOSED ALGORITHM

Using HNB as a base classifier, we constructed an algorithm under the AL scheme, named Active Hidden Naive Bayes. The advantages of HNB in terms of speed and robustness, along with the improvements compared to NB, makes it a great candidate to create performant models with very few instances. We used Pool-Based Scenario and Uncertainty Sampling using HNB for ranking the instances of the U subset and later classifying the test ones. As it considers the ranking process, a numerous of methods are suggested [7]. For our algorithm we used Entropy method which is suitable for both binary and multiclass problems, defined by the next equation:

$$x_{Ent} = \arg \max_{x} - \sum_{y} P_\theta(y|x) \log P_\theta(y|x) \quad (2)$$

The stopping criterion is determined by the Budget Plan (B) and it is defined of as a fix percentage (b) of instances belonging to U that got labeled by the Human Expert (HE). The algorithm will stop when the size of L reaches B. This way, the budget plan is fixed to the size of the U as it is demonstrated into the Algorithm Section.

The following algorithm demonstrates the procedure of the learning phase, given a HE and L, U subsets:

In each iteration, we discretized a number of numeric attributes of L into nominal using the information entropy minimization heuristic, as described in [16], with the default options. The trained HNB on L of the last iteration is used to predict the labels in the test set. In the last task, HNB thrives again as it is very efficient in classification time that bounds in polynomial complexity as well [4].

Algorithm 1 Active Hidden Naive Bayes (Input - Main Body - Output)

HE Human Expert
L (U) Initial (Un)Labeled set
Cl Base Classifier → Hidden Naive Bayes
QS Query Strategy → Entropy Sampling
b Budget Plan Percentage
B Budget Plan → size(L) + b * size(U)
TR Top Rated instances → empty
while size(L) is smaller than B **do**
(Re)train CL over current Discretized L
  Assign uncertainty values on U based on QS(CL)
  Add *top rated* instances in TR
Remove TR from U
Ask HE for annotating instances into TR
Merge TR with current L and then create an empty TR
Output: Predict labels of test set using the CL *trained on finally formatted* L

4 EXPERIMENTS AND RESULTS

4.1 Datasets

For our we selected a number of 14 datasets from UCI repository with good diversity in terms of attributes and size, with both binary and multiclass problems. In Table 1 is described the formulation of each dataset as the number (#) of instances, classes and attributes combined with the number of the type (Categorical or Numerical) of each instance. The last column describes the percentage ratio of the class with the most instances (Majority class) and the class with the least instances (Minority class) compared to the rest of the instances.

4.2 Experimental procedure

To begin with, we define the Labeled Ratio (R) as the percentage ratio of the size of L compared to the size of L and U combined and it is described with the following formula:

$$R = \frac{L}{L+U} * 100 \quad (3)$$

We used 3-fold validation, meaning that our learner is trained at the 2/3 of the total instances of each dataset and the rest 1/3 is used for testing. Each train set is separated to initial L and U subsets, according to the R-percentage. For our experiments we choose $R = 10\%, 20\%$ and we set the b-percentage to $b = 10\%$ in order to restrict highly the size of L and the participation of HE.

To compare with, we selected four well-known state-of-the-art classifiers and we trained them under the same scheme. The selected classifiers are Sequential Minimal Optimization (SMO) from SVM classifiers, Naive Bayes (NB), C4.5 Decision Tree implemented in JAVA (J48) and Instance Base K-nearest-neighbor. All of them have been used with their default parameters as introduced in WEKA tool, where $K = 5$ for the latter one (5NN).

Table 1: Formulation of datasets

Dataset	# of instances	# of classes	# of attributes	Categorical/Numerical attributes	Majority/Minority Class %
Car	1728	4	6	6 / 0	70,023 / 7,755
Chess	3196	2	36	36 / 0	52,222 / 47,778
credit-g	1000	2	20	13 / 7	70,0 / 30,0
Flare	1066	6	11	11 / 0	31,051 / 12,946
German	1000	2	20	13 / 7	70,0 / 30,0
kr-vs-kp	3196	2	36	36 / 0	52,222 / 47,778
mammographic	830	2	5	0 / 5	51,446 / 48,554
Optdigits	5620	10	64	0 / 64	10,178 / 19,716
Ring	7400	2	20	0 / 20	50,486 / 49,514
Segment	2310	7	19	0 / 19	14,268 / 28,572
Spambase	4579	2	57	0 / 57	60,583 / 39,417
Texture	5500	11	40	0 / 40	9,091 / 18,182
Twonorm	7400	2	20	0 / 20	50,041 / 49,959
Wisconsin	683	2	9	0 / 9	65,007 / 34,993

Table 2: Prediction results for R = 10% (left) and R = 20% (right)

Dataset	HNB (Ent)	SMO (Ent)	NB (Ent)	J48 (Ent)	5NN (Ent)	HNB (Rnd)	SMO (Rnd)	HNB (Ent)	SMO (Ent)	NB (Ent)	J48 (Ent)	5NN (Ent)	HNB (Rnd)	SMO (Rnd)
Car	**91,38**	89,70	83,62	75,93	84,66	85,01	87,21	**92,36**	91,09	87,62	84,09	87,44	87,33	87,44
Chess	**97,40**	96,65	94,84	95,15	88,49	89,49	94,24	97,09	97,00	94,37	**97,87**	92,08	91,52	94,74
credit-g	72,70	67,91	**74,90**	70,00	70,30	70,70	68,30	70,90	**73,00**	70,60	69,20	69,70	71,20	69,50
Flare	73,73	74,20	**74,77**	66,98	70,64	71,76	72,42	**74,39**	72,61	74,11	70,17	73,82	70,73	72,61
German	71,61	71,30	**72,80**	71,00	70,00	67,80	70,60	71,50	71,00	**74,10**	69,50	70,00	71,40	71,70
kr-vs-kp	**98,06**	96,59	96,06	94,37	87,52	89,39	94,15	97,18	97,15	92,87	**97,72**	91,65	91,61	94,52
Mammographic	78,07	80,36	**81,44**	74,71	69,40	79,88	79,40	82,77	81,81	79,64	82,18	72,17	**83,25**	82,05
Optdigits	94,89	97,58	91,73	80,66	**98,10**	93,75	96,33	96,05	97,83	91,57	83,63	**98,40**	94,23	96,90
Ring	94,93	75,78	**97,92**	83,03	68,80	94,88	76,20	95,65	76,05	**97,92**	87,14	66,05	95,72	75,26
Segment	**93,68**	90,35	86,28	89,09	90,82	91,08	87,19	**94,59**	91,21	86,23	92,38	91,43	93,16	90,22
Spambase	**93,60**	91,54	81,92	90,30	80,68	91,04	89,58	**93,32**	91,56	80,40	90,60	82,92	91,80	89,93
Texture	93,95	**98,45**	81,91	86,04	96,58	89,89	96,07	95,64	**98,38**	80,11	88,73	97,62	92,07	97,11
Twonorm	94,84	97,77	**97,82**	78,92	95,34	95,91	97,64	96,59	**97,74**	97,73	82,61	96,31	96,68	97,65
Wisconsin	**96,78**	96,63	96,19	92,83	**96,78**	96,34	**96,78**	96,63	96,78	96,48	92,68	**97,22**	95,76	96,78
Average	**88,97**	87,49	86,59	82,07	83,44	86,21	86,15	**89,62**	88,09	85,98	84,89	84,77	87,60	86,89

Additionally, we ran these experiments under Random Sampling as well, a non-sophisticated strategy that selects the instances for labeling in a random manner, in order to demonstrate the superiority of Uncertainty Sampling strategy. The JCLAL framework was exploited for conducting the described AL part of our experiments which is compatible with the WEKA tool [17].

4.3 Results

In the following table is demonstrated the prediction results for each algorithm for both $R = 10\%$ and $R = 20\%$ for Entropy (Ent) and Random (Rnd) Sampling. Due to lack of space, we only included the HNB and SMO classifiers from random sampling results, as these classifiers showed the best predictive behavior under this Query Strategy. The best prediction accuracy is annotated in bold format per dataset/line. In the last row, the average accuracy is demonstrated.

According to the results, the proposed algorithm outperformed its rivals in terms of average prediction accuracy for the examined datasets. In order to further verify its effectiveness, we applied a statistical analysis using the non-parametrical Friedman test, a popular method that compares the performance of the algorithms by examining if the null hypothesis about their similarity holds [18]. On Table 3 the Freidman Ranking scores are demonstrated for $R = 10\%$ and $R = 20\%$ regarding results of Table 2. From this kind of evaluation, we observe that the proposed algorithm achieved the best ranking for both labeled ratio rates, managing to improve its performance when it was provided with more initial labeled data for the majority of the examined problems. Regarding the QS

Table 3: Friedman ranking for R = 10% (left) and R = 20% (right)

Algorithm	Ranking	Algorithm	Ranking
HNB (Ent)	**2,500**	**HNB (Ent)**	**2,571**
SMO (Ent)	2,929	SMO (Ent)	3,071
NB (Ent)	3,643	SMO (Rnd)	4,107
SMO (Rnd)	4,214	HNB (Rnd)	4,214
HNB (Rnd)	4,357	NB (Ent)	4,357
5NN (Ent)	4,929	5NN (Ent)	4,607
J48 (Ent)	5,429	J48 (Ent)	5,071

comparison, *Ent* with the use of HNB outperformed both HNB and SMO under *Rnd*.

5 CONCLUSIONS

To sum up, the HNB learner is exploited under AL scheme in this work obtaining promising results in terms of prediction accuracy and time efficacy. Compared to four state-of-the-art classifiers under the same scheme, its superiority performance is demonstrated in both average accuracy and statistical analysis. As future work, we should measure the performance of HNB on larger datasets comparing it to Neural Networks approaches based on few shot learning and investigate its behavior under more realistic scenarios like noisy human oracles [19].

REFERENCES

[1] M Wiggins, Ashraf Saad, Brian Litt and George Vachtsevanos. 2008. Evolving a Bayesian classifier for ECG-based age classification in medical applications. Applied Soft Computing, Vol. 8, (January 2008), 599–608, https://doi.org/10.1016/j.asoc.2007.03.009

[2] Nir Friedman, Dan Geiger and Moises Goldszmidt. 1997. Bayesian Network Classifiers. Machine Learning 29, (November 1997), 131–163, https://doi.org/10.1023/A:1007465528199

[3] David M. Chickering. 1996. Learning Bayesian Networks is NP-Complete. In: Fisher D., Lenz HJ. (eds) Learning from Data. Lecture Notes in Statistics, Vol. 112. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-2404-4_12

[4] Liangxiao Jiang, Harry Zhang and Zhihua Cai. 2009. A Novel Bayes Model: Hidden Naive Bayes. IEEE Transactions on Knowledge and Data Engineering 10, Vol. 21, (October 2009), 1361-1371, https://doi.org/10.1109/TKDE.2008.234

[5] Nikos Fazakis, Stamatis Karlos, Sotiris Kotsiantis and Kyrgiakos Sgarbas. 2016. Self-labeled Hidden Naive Bayes algorithm for semi-supervised classification. 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA), (December 2016), 1-6, https://doi.org/10.1109/IISA.2016.7785414

[6] Friedhelm Schwenkera and EdmondoTrentinb. 2014. Pattern classification and clustering: A review of partially supervised learning approaches. Pattern Recognition Letters, Vol. 37, (February 2014), 4-14, https://doi.org/10.1016/j.patrec.2013.10.017

[7] Burr Settles. 2012. Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 1, Vol. 6, (June 2012), 1-114, https://doi.org/10.2200/S00429ED1V01Y201207AIM018

[8] H. M. Sajjad Hossain, Nirmalya Roy and Md. Abdullah Al Hafiz Khan. 2017. Active learning enabled activity recognition. Pervasive and Mobile Computing 2, Vol. 38, (July 2017), 312-330, https://doi.org/10.1016/j.pmcj.2016.08.017

[9] Andrew McCallum and Kamal Nigam. 1998. Employing EM and Pool-Based Active Learning for Text Classification. ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, (July 1998), 350–358

[10] Vangjel Kazllarof, Stamatis Karlos, Sotiris Kotsiantis and Michalis Xenos. 2017. Automated hand gesture recognition exploiting Active Learning methods. PCI 2017: Proceedings of the 21st Pan-Hellenic Conference on Informatics, (September 2017), Article 3, 1-6, https://doi.org/10.1145/3139367.3139414

[11] Burr Settles. 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, (January 2009), http://digital.library.wisc.edu/1793/60660

[12] Xin Li and Yuhong Guo. 2013. Active Learning with Multi-Label SVM Classification. IJCAI '13: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, (August 2013), 1479-1485

[13] Xuchun Li, Lei Wang and Eric Sung. 2004. Multilabel SVM active learning for image classification. ICIP '04: International Conference on Image Processing, Vol. 4, (October 2004), 2207-2210, https://doi.org/10.1109/ICIP.2004.1421535

[14] Steven C. H. Hoi, Rong Jin, Jianke Zhu and Michael R. Lyu. 2009. Semisupervised SVM batch mode active learning with applications to image retrieval. ACM Trans. Inf. Syst. 27, 3, Article 16 (May 2009), 29 pages, https://doi.org/10.1145/1508850.1508854

[15] Vangjel Kazllarof, Stamatis Karlos and Sotiris Kotsiantis. 2019. Active learning Rotation Forest for multiclass classification. Computational Intelligence 4, Vol. 35, (May 2019), 891-918, https://doi.org/10.1111/coin.12217

[16] Usama M. Fayyad and Keki B. Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. Thirteenth International Joint Conference on Articial Intelligence, Vol. 2, Morgan Kaufmann Publishers, (September 1993), 1022-1027

[17] Oscar Reyes, Eduardo Pérez, María del Carmen Rodríguez-Hernández, Habib M. Fardoun and Sebastián Ventura. 2016. JCLAL: A Java Framework for Active Learning. Journal of Machine Learning Research, Vol. 17, (May 2016), 1-5

[18] Rob Eisinga, Tom Heskes, Ben Pelzer, Manfred Te Grotenhuis, Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers, BMC Bioinformatics. 18 (2017) 1–18. https://doi.org/10.1186/s12859-017-1486-2

[19] Gaurav Gupta, Anit Kumar Sahu, Wan-Yi Lin, Learning in Confusion: Batch Active Learning with Noisy Oracle, (2019). http://arxiv.org/abs/1909.12473