A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features

Ali Haghpanah jahromi
School of Electrical and Computer Engineering
Shiraz University
Shiraz,Iran
Email: alihgpj@gmail.com

Mohammad Taheri
School of Electrical and Computer Engineering
Shiraz University
Shiraz,Iran
Email: motaheri@shirazu.ac.ir

***Abstract*— The naive Bayes is one of the useful classification techniques in data mining and machine learning. Although naive Bayes learners are efficient, they suffer from the weak assumption of conditional independence between the attributes. Many algorithms have been proposed to improve the effectiveness of naive Bayes classifier by inserting discriminant approaches into its generative structure. Combining generative and discriminative viewpoints is done in many algorithms e.g. by use of attribute weighting, instance weighting or ensemble method. In this paper, a new ensemble of Gaussian naive Bayes classifiers is proposed based on the mixture of Gaussian distributions formed on less conditional dependent features extracted by local PCA. A semi-AdaBoost approach is used for dynamic adaptation of distributions considering misclassified instances. The proposed method has been evaluated and compared with the related work on 12 UCI machine learning datasets and achievements show significant improvement on the performance.**

***Index Terms*— ensemble naive Bayes; local PCA; multi-modal classification; Gaussian naive Bayes.**

I. Introduction

Naive Bayes classifier, simply naive Bayes, is an efficient classifier that is one of the top 10 algorithms in data mining [1]. Naive Bayes is a useful classifier that is used widely in many applications such as: text categorization [2], document judgment (e.g. spam filtering [3]) and data stream classification [4]. Naive Bayes is a generative model based classifier [5] with a fast learning and testing process.

Bayesian classifiers, work based on the Bayesian rule and probability theorems. It has been proven that learning an optimal Bayesian classifier from training data is an NP-hard problem [6]. A simplified version of Bayesian classifier called as naive Bayes uses two assumptions. The former is that, given the class label, attributes are conditionally independent and the latter is that, no latent attribute affects on the label prediction process [7].

Assume, the vector $(x_1, ..., x_{\mathrm{n}})$ represents the n attributes of the instance x. Let c represents the class label of the instance x. The probability of observing x given the class label c can be computed by Equ.(1) that is relaxed by above assumptions.

$$p(x_1, ..., x_{\mathrm{n}}|c) = \prod_{i=1}^{n} p(x_{\mathrm{i}}|c) \tag{1}$$

In order to predict the label of instance x, the probability of instance x in each class label is computed. The class with the maximum probability is identify as the class label of the instance x. Equation (2) defines the label estimation process of instance x.

$$C(x)_{\mathrm{NB}} = \underbrace{argmax}_{c} \quad p(c) \prod_{i=1}^{n} p(x_{\mathrm{i}}|c) \tag{2}$$

The conditional independence assumption between the attributes in naive Bayes is weak and rarely correct in most of the real problems except of situations in which, the attributes are extracted from independent processes. Some methods have been introduced for improving the conditional independence assumption in naive Bayes.

Gaussian naive Bayes classification is a case of naive Bayes method with an assumption of having a Gaussian distribution on attribute values given the class label. For example, suppose that i$^{\mathrm{th}}$ attribute is continuous and its mean and variance are represented by $\mu_{c,i}$ and $\sigma^2_{c,i}$, respectively, given the class label c. Hence, the probability of observing the value x_{i} in i$^{\mathrm{th}}$ attribute given the class label c, is computed by Equ.(3) that is also called as normal distribution.

$$p(x_i|c) = \frac{1}{\sqrt{2\pi\sigma^2_{c,i}}} \exp^{(-\frac{(x_i-\mu_{c,i})^2}{2\sigma^2_{c,i}})} \tag{3}$$

Even with Gaussian estimation, this method suffers the weakness of conditional independence of attributes.

In section II, related work has been investigated. In section III, the proposed method and the objective function is explained. Experimental results have been shown and discussed in section IV and finally, the paper has been concluded in section V.

II. Related work

Many algorithms have been proposed with different viewpoints to improving naive Bayes classifiers. For reforming the weak conditional independence assumption, some algorithms parameterized the method (e.g. by attribute weighting) or extended the structure.

978-1-5386-2585-9/17/$31.00 ©2017 IEEE

For example, in *augmented naive Bayes*, dependency of each attribute has been extended to some attributes in addition of the class label [8]. In other algorithms, attributes can be weighted differently to adjust the effect of each attribute. But, the conditional independence is yet assumed. This algorithm have been combined the kernel methods with the weighting methods [9].

In the proposed method explained in section III, usage of the principle component analysis (PCA) for improving the conditional independence assumption is discussed in the case of assuming the Gaussian distribution.

Ensemble methods are also related to the proposed method that uses a semi-AdaBoost approach in the learning process. An ensemble method combines some weak learners to achieve a better performance than the base models [10]. In order to predict the label of the instances, ensemble method may use a weighted voting between the classifiers [11]. AdaBoost is one of the ensemble learning methods in which, some base classifiers are learned one by one. Due to the learning a new classifier, weights of misclassified instances in the earlier learners increases to pay more attention to learn their distribution as much as possible. Finally, each learner is also weighted based on its performance on the training data. In the proposed ensemble approach, after each phase of learning, a new component is constructed to handle the misclassified instances.

III. The Proposed Method

The proposed method, as an ensemble Gaussian naive Bayes classifier is explained in following. In this method, distribution of each class label is estimated considering only the training instances of the associated class separately. The distribution is assumed to be Gaussian. In this model, probability of observing an instance x given the class label c is computed by use of equations Equ.(2) and Equ.(3). However, due to improve the conditional independency of the features, instances of each class are mapped to the new coordination system described by the new features extracted from PCA without any reduction.

The main reason of this transformation is that, probability of the instances along these features are independent. In other words, the probability of attribute values are independent based on these features if the probability of an attribute value is explained only by the mean and variance of training instances along that attribute. This claim is shown in later discussions. However, instances of a class may be distributed in many models. It means that, a mixture of above Gaussian models may be required to model a class.

At the first iteration of the proposed algorithm, instances of each class is considered as a cluster. In the next step, in each cluster, a local PCA is applied separately to extract the features of that cluster. Then, a Gaussian model is formed on the features of each cluster. The probability of an instance in each Gaussian is computed by Equ.(3) considering the associated features of that cluster and the class label with the maximum probability is assigned to the given instance.

In each of the next iterations, all training instances are classified by the constructed model of the previous iteration. Then, misclassified instances of each class label is assigned to a new cluster. Afterwards, the same process is done on each (old and new) cluster to extract local features and generating the Gaussian models. The clusters with low rank covariance matrix to apply the PCA are ignored. The process is iteratively repeated until no new cluster is generated or the training classification error decreases respect to the previous iteration. The flowchart of the proposed method is given in Fig.1.

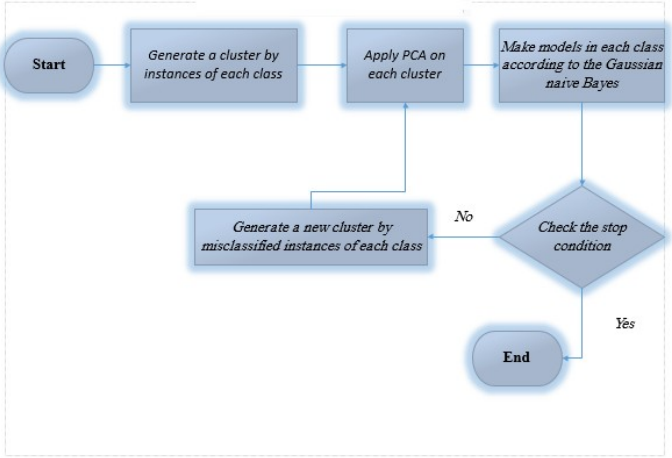

Fig. 1. The flowchart of the proposed classifier

In subsection III-A, it is explained that, why use of PCA can improve the weak conditional independence assumption of Gaussian naive Bayes. In subsection III-B, the ensemble method of constructed models in each class for label prediction process is expressed.

A. Improving the conditional independence assumption

As explained, the Gaussian naive Bayes is used for estimating the probability of instances in each class. Gaussian distribution is quite well for estimation distribution when the data is continuous and large enough. In following, it is shown that Gaussian distribution on attributes extracted by PCA, improves the conditional independence of attributes.

PCA extracts orthogonal dimensions with maximum variance of training instances along them. In addition, if the data is independently distributed with a normal distribution along orthogonal dimensions, these dimensions can be extracted by PCA if there are enough number of instances. Moreover, with the above assumption about the distribution, value of j^{th} attribute comes from a normal distribution with a given mean, regardless of the values of other attributes. As shown in Fig.2,

principle components of distribution (Eigen vectors of covariance matrix) have been extracted along which, the variance of instances are maximum. If an instance is represented by its original features, by increasing the value of one attribute, the mean of distribution of other attribute also increases. However, if the instances are represented by red basis, the mean of distribution of each attribute is independent of the value of other one.

As shown in Fig.2, instances have been distributed along red directions. In this case, the probability of observing green triangles is more than blue ones because, variance along the main direction is more and consequently, the probability based on Equ.(3) is higher.

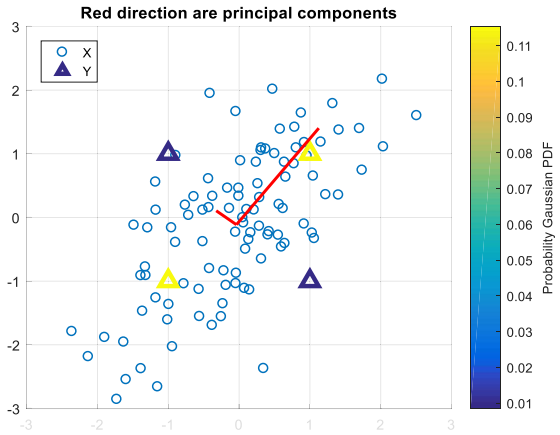

Fig. 2. A sample of local PCA on training instances of a class

B. Ensemble method

Due to have more than one clusters in some classes, there are two approaches of computing the class label of each instance. The former one which is described earlier, is called as single winner. In this approach, the class of the cluster with maximum probability is assigned to the instance. The latter one is called as weighted vote. In this approach, probability of belonging an instance to a class is the sum of membership probability of the instance to all clusters of that class. Finally, the class with maximum membership probability is the winner class.

Membership probability of an instance x to j^{th} cluster of k^{th}class (i.e. represented as c_{kj}) is presented by Equ.(4).

$$p(x|c_{kj}) = \prod_{i=1}^{n} p(x_i|c_{kj}) \tag{4}$$

After computing the probability instance x in each cluster separately, the accumulated membership probability of instance x in each class is computed by Equ.(5) assuming the same prior probability for each cluster.

$$p(x|c_k) = \sum_{j=1}^{l} p(x|c_{kj}) \tag{5}$$

Final class label of the instance is computed by Equ.(2) ignoring the prior probability of the class labels. In this ensemble method, no prior probability was considered as the weight of each cluster but for weighted versions, various approaches such as large margin based component weighting can be used [12].

Finally, it should be underlined that, the main contribution of this paper is combination of generative and discriminative views. In discriminative view, the force is on increasing the classification accuracy by separating misclassified samples in a new cluster. From the generative point of view, the probability density function of each cluster is generated independently and only based on the distribution of the associated samples.

IV. Experimental Results

In this section, the proposed method has been evaluated and compared with 6 strong classifiers:

- Multilayer perceptron(MLP),
- Logistic regression classifier(LOG),
- Bayesian network(BN),
- Naive Bayes(NB),
- K-nearest neighbors(KNN),
- Linear support vector machine(SVM).

For more suitable evaluation, Weka software and LIBSVM [13] have been used. Moreover, 10-times 10-flods cross-validation has been used to evaluate the methods and significance of results have been tested by T-Test between the proposed method and others. It is tried to find the optimum values of parameters in each method e.g. Learning rate in multilayer perceptron, k in k-nearest neighbors and C in linear SVM have been adjusted by 10-folds cross validation on training data.

TABLE I
Specification of UCI datasets used in this paper

Dataset name	num feature	num instance	num class
Iris	4	150	3
banknote authentication	5	1372	2
Blood Transfusion Service Center	5	748	2
EEG Eye State	15	14980	2
Seeds	7	210	3
Connectionist Bench	60	208	2
User Knowledge Modeling	5	403	4
Blogger	6	100	2
Parkinsons	23	197	2
Wine	13	178	3
Balance Scale	4	625	3
Statlog (Vehicle Silhouettes)	18	946	4

TABLE II
COMPARISON OF ACCURACY OF THE PROPOSED METHOD WITH 6 STRONG CLASSIFIERS BY 10-TIMES 10-FOLDS CROSS VALIDATION

Dataset	MLP	LOG	BN	NB	KNN	SVM	Proposed method	#cluster
Iris	96.6 ± 0.49	97.26 ± 0.85	92.86 ± 0.70	95.46 ± 0.42	95.6 ± 0.56	98.13 ± 0.52	97.13 ± 0.83	**1**
T-test	*null*	*null*	*reject*	*reject*	*reject*	*null*	-	-
banknote	99.73 ± 0.31	98.97 ± 0.10	94.69 ± 0.60	83.96 ± 0.11	99.78 ± 0.02	99.14 ± 0.04	98.59 ± 0.08	**1**
T-test	*reject*	*reject*	*reject*	*reject*	*reject*	*reject*	-	-
Blood Transfusion	78.04 ± 0.52	77.01 ± 0.15	74.02 ± 0.69	75.17 ± 0.20	76.31 ± 0.94	74.32 ± 2.23	76.84 ± 0.32	**1.845**
T-test	*reject*	*null*	*reject*	*reject*	*null*	*reject*	-	-
EEG Eye State	55.65 ± 0.33	59.64 ± 0.21	76.86 ± 0.24	47.08 ± 0.34	83.93 ± 0.61	64.07 ± 0.17	78.23 ± 0.16	**3.6**
T-test	*reject*	*reject*	*reject*	*reject*	*reject*	*reject*	-	-
Seeds	94.09 ± 1.19	95.28 ± 0.93	90.19 ± 0.64	90.38 ± 0.37	93.42 ± 0.99	96.33 ± 0.89	94.23 ± 0.57	**1.46**
T-test	*null*	*null*	*reject*	*reject*	*null*	*reject*	-	-
Connectionist	81.75 ± 2.11	74.65 ± 1.36	76.53 ± 2.13	67.32 ± 1.07	85.77 ± 0.66	78.46 ± 1.58	79.75 ± 1.29	**1**
T-test	*null*	*reject*	*reject*	*reject*	*reject*	*null*	-	-
User Knowledge	92.35 ± 1.61	93.69 ± 0.40	83.52 ± 1.18	88.82 ± 0.51	81.88 ± 1.13	93.90 ± 0.63	94.56 ± 0.48	**1.02**
T-test	*reject*	*null*	*reject*	*reject*	*reject*	null	-	-
Blogger	78.30 ± 2.83	73.00 ± 1.15	79.80 ± 2.20	71.50 ± 1.43	82.90 ± 1.52	69.70 ± 2.98	82.80 ± 2.89	**1**
T-test	*reject*	*reject*	*null*	*reject*	*null*	*reject*	-	-
Balance Scale	90.33 ± 0.80	89.21 ± 0.59	69.23 ± 0.88	90.53 ± 0.25	89.66 ± 0.38	91.67 ± 0.007	91.69 ± 0.55	**1.466**
T-test	*reject*	*reject*	*reject*	*reject*	*reject*	*null*	-	-
Parkinsons	88.57 ± 1.97	86.03 ± 1.09	86.99 ± 1.49	69.27 ± 0.72	93.52 ± 0.57	86.60 ± 0.56	85.84 ± 0.96	**1.11**
T-test	*reject*	*null*	*null*	*reject*	*reject*	*null*	-	-
Wine	97.24 ± 0.56	97.23 ± 0.56	97.35 ± 1.14	97.74 ± 0.36	95.25 ± 0.54	95.50 ± 0.57	97.69 ± 0.49	**1**
T-test	*null*	*null*	*null*	*null*	*reject*	*reject*	-	-
Statlog	81.12 ± 1.21	80.005 ± 0.21	71.38 ± 0.72	44.62 ± 0.64	71.51 ± 0.38	80.44 ± 0.58	84.23 ± 0.42	**1.01**
T-test	*reject*	*reject*	*reject*	*reject*	*reject*	*reject*	-	-
#Better	**5**	**5**	**9**	**11**	**5**	**5**	-	-
#Worse	**3**	**1**	**0**	**0**	**4**	**2**	-	-
#No Diff.	**4**	**6**	**3**	**1**	**3**	**5**	-	-

Methods have been compared on 12 UCI datasets listed in Table I and the results have been reported in Table II. In Table II, the column named as #cluster, is the average number of clusters made in each class in each training phase. The results of T-Test between the proposed method and each other method have been reported in the next row of associated method. The word null represents the null hypothesis (there is no significant difference) and the word reject represents the significance of the difference between the methods. The last rows show the number of datasets on which the proposed method have significantly better or worse performance or without significant difference, respectively.

Based on the results, just K-NN and MLP with their best parameters (not in hand in many situations) can, to some extent, compete with the proposed method. It should be underlined again that, the proposed method has achieved a good performance without any parameter tuning and even use of prior probability of classes or clusters.

Experimental results show good performance of the proposed algorithm. The main reason of this improvement may be the proper distribution of classes, near to Gaussian distribution, that may be a weak assumption in some datasets.

V. CONCLUSION

In this paper, a non-parametric semi-AdaBoost ensemble and hybrid of generative and discriminative approaches has been proposed based on local PCA on one or more modalities of each class. Experimental results, show significance improvement in many cases of comparisons. As the future work, taking some parameters such as prior probability of classes into consideration may improve the performance of the proposed method. Weighting based on large margin approaches, rule extraction, stability on imbalanced or noisy datasets can form other research fields of the future.

REFERENCES

[1] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[2] L. Jiang, Z. Cai, H. Zhang, and D. Wang, "Naive bayes text classifiers: a locally weighted learning approach," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 2, pp. 273–286, 2013.

[3] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive bayes algorithm for spam filtering," in *Performance Computing and Communications Conference (IPCCC), 2016 IEEE 35th International*. IEEE, 2016, pp. 1–8.

[4] C. Hue, M. Boullé, and V. Lemaire, "Online learning of a weighted selective naive bayes classifier with non-convex optimization," in *Advances in Knowledge Discovery and Management*. Springer, 2017, pp. 3–17.

[5] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Advances in neural information processing systems*, vol. 2, pp. 841–848, 2002.

[6] D. M. Chickering, D. Geiger, D. Heckerman *et al.*, "Learning bayesian networks is np-hard," Technical Report MSR-TR-94-17, Microsoft Research, Tech. Rep., 1994.

[7] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[8] E. J. Keogh and M. J. Pazzani, "Learning the structure of augmented bayesian classifiers," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 04, pp. 587–601, 2002.

[9] Z.-L. Xiang, X.-R. Yu, and D.-K. Kang, "Experimental analysis of naïve bayes classifier based on an attribute weighting framework with smooth kernel density estimations," *Applied Intelligence*, vol. 44, no. 3, pp. 611–620, 2016.

[10] Z.-H. Zhou, "Ensemble learning," *Encyclopedia of biometrics*, pp. 411–416, 2015.

[11] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[12] M. Taheri, H. Azad, K. Ziarati, and R. Sanaye, "A quadratic margin-based model for weighting fuzzy classification rules inspired by support vector machines," *Iranian Journal of Fuzzy Systems*, vol. 10, no. 4, pp. 41–55, 2013.

[13] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.