International Journal of Engineering & Technology, 7 (4.11) (2018) 126-132

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Partial Histogram Bayes Learning Algorithm for Classification Applications

Haider O. Lawend, Anuar M. Muad*, Aini Hussain

Center for Integrated Systems Engineering and Advanced Technologies (INTEGRA), Faculty of Engineering and Built Environment
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
**Corresponding author E-mail: anuar_muad@ukm.edu.my*

Abstract

This paper presents a proposed supervised classification technique namely partial histogram Bayes (PHBayes) learning algorithm. Conventional classifier based on Gaussian function has limitation when dealing with different probability distribution functions and requires large memory for large number of instance. Alternatively, histogram based classifiers are flexible for different probability density function. The aims of PHBayes are to handle large number of instances in datasets with lesser memory requirement, and fast in training and testing phases. The PHBayes depends on portion of the observed histogram that is similar to the probability density function. PHBayes was analyzed using synthetic and real data. Several factors affecting classification accuracy were considered. The PHBayes was compared with other established classifiers and demonstrated higher accurate classification, lesser memory even when dealing with large number of instance, and faster in training and testing phases.

Keywords: *Classification; Histogram noise estimation and reduction; Histogram probability distribution; Naïve Bayes; Supervised learning*.

1. Introduction

Large databases that are dynamically changing in which new information and classes are constantly added poses great challenge to many machine learning algorithms to perform classification tasks. This situation always occurs in many applications like image and data classification, robotics, and many others. Recent advances in the robotic technology requires not only fast and accurate classifier, but also a classifier that can learn newly added information. However, with the abundant quantity of information that the robot need to learn, it may hinder the learning progress of the robot [1]. Applications that involve a stream of data have a tendency to expand the size of its database and this may reduce the accuracy and learning rate of the classifier [2].

Classifiers like artificial neural network (ANN) depends on the complexity of its structure, such as number of nodes, hidden layers, and activation function [3] to deal with large databases. The drawback of the ANN is that every time new information or classes are added, retraining is required, which can slow down the learning capabilities of the ANN [4]. Therefore, ANN seems to have serious limitation for large databases classification.

A distance based classifier like supper vector machine (SVM) needs to learn all the support vectors using two different strategies: one versus one and one versus rest. For large databases, the number of support vectors are large, which make the learning phase of the SVM as a very time consuming process [5-8]. Moreover, with new addition of information and classes, the SVM need to be retrained. For the distance based classifier, 1st Nearest Neighbour (1stNN) can produce high accurate classification even for large number of instance [9-11]. The disadvantage of 1stNN is that it suffers from curse of dimensionality, especially in applications involving large number of instance because the computationally effort is expensive [12]. The 1stNN keeps s number of training instance, and each instance contains n features. Therefore, the memory requirement for the 1stNN is only sn. The improvement of the 1stNN have been reported in many works [11-16] to enhance the accuracy and speed of classification. The speed during testing phase of the 1stNN can be improved by considering multiple prototype of classes as in Nearest Subclass Classifier (NSC) [10]. In general, to find the prototype, a clustering technique like k-means can be used. Nearest Class Mean (NCM) is a type of NSC but with only one prototype to represent a class probability density. In comparison, NSC is faster than 1stNN during the testing phase, but slower during the training phase. NCM is generally fast during training and testing phases because it only calculates the a testing instance and the prototype [5, 10, 12]. NCM tends to generalize well for the base configuration [17]. If the class probability density is spread wider, the accuracy of NCM deteriorates. The memory capacity of the NSC consists of sn and subclass means, mn, where m is the number of subclass or prototype. Since NCM only measure the nearest class mean to a testing instance, the NCM memory is only n, which is smaller than that of the 1stNN and NSC.

Classifiers that are based on Bayesian classification like a simple Naïve Bayes (NB) represents a class feature probability density using only a single Gaussian function [18]. NB assumes that the distribution of the class density is Gaussian, thus allowing fast and accurate classification. However, if the distribution is non-Gaussian [9], the accuracy of NB decreases. The memory capacity of the NB is double than the NCM because NB needs to keep information about the class mean and variance. Gaussian Mixture Model Classifier (GMMC) represents the class feature probability density using mixture of Gaussian. Expectation maximization (EM) is used to estimate the parameters of the GMMC [19]. As the EM requires iterative procedure, the training phase of the GMMC is relatively slow [20-23]. Flexible Naïve Bayes (FNB) represents the class feature probability density using kernel density estimation [24-25]. The most widely used kernel is a Gaussian kernel [26]. FNB that uses Gaussian kernel is similar to GMMC because both approach use multiple Gaussian as a representation of a class feature probability density. Both FNB and GMMC are slower than NB but their accuracy is higher than that of the NB. Because GMMC involves of mixture of Gaussian, its memory capacity consists of all the training instances, sn, the class means, mn, and the variances, mn. Altogether the memory for the GMMC is n(s+2m), which is larger than the previously mentioned classifiers.

For applications involving large databases, ANN, SVM, and 1stNN, NSC, and NCM are considered as slow techniques. Bayesian based technique like NB, GMMC, and FNB are more suitable. In a previous work [27], our proposed PHBayes demonstrated faster and more accurate classification compared to NB, GMMC, 1stNN, and NCM. PHBayes need not to keep the number of instance in its memory, but it requires a memory space for the probability density of the observed histogram bins. Further analyses on the impact of PHBayes with several factors affecting the classification accuracy are investigated in this paper. Section 2 provides a description about Naïve Bayes classifier that was used as a basis for the PHBayes, Section 3 explains our technique, supervised PHBayes classifier, Section 4 presents results and discussion, finally Section 5 concludes the finding of this paper.

2. Naïve Bayes Classifier

Bayesian classifier is an algorithm which uses the Bayesian rule as the base of the classifier. Class feature probability density is represented as parameters. All the features are assumed to be independent with one another. Suppose that **X** = (X_1,…,X_n) is an instance vector containing n features. The conditional probability p of class C in the learning system is given in (1).

$$p(C\,|\,\mathbf{X})=\frac{p(C)\,p(\mathbf{X}|C)}{p(\mathbf{X})} \qquad (1)$$

Instance **X** is classified into class C which has the highest posterior probability. The conditional probability of the NB can be further written as in (2).

$$p(C\,|\,\mathbf{X})=p(C)\prod_{i=1}^{n}p(X_i\,|\,C) \qquad (2)$$

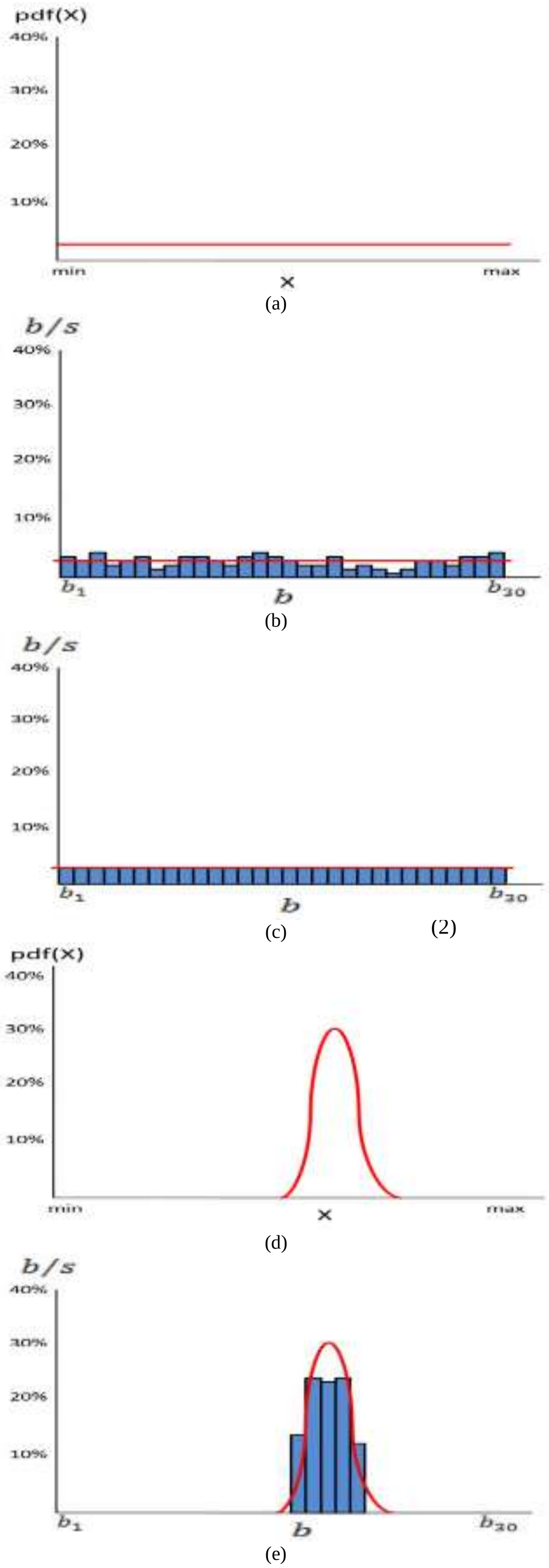
For continuous features, Gaussian distribution is generally used, while for discrete features, Direchlet distribution is typically employed [28]. In some cases, each nominal feature is converted into a binary feature [24]. A single Gaussian probability density distribution function is provided in (3).

$$p(X_i\,|\,C)=\frac{1}{\sqrt{2\pi\Sigma_i}}e^{-\frac{(X_i-\mu_i)^2}{2\Sigma_i}}=N(X_i,\mu_i,\Sigma_i) \qquad (3)$$

where μ is the mean and ∑ is the variance, and X is the input. Due to its simplicity, speed and accuracy in large databases, NB has become a popular classifier in many applications [24, 28]. The accuracy of the NB classifier is comparable to the ANN [29] but lesser than that of the SVM [30-32].

3. Supervised Partial Histogram Bayes

The accuracy of the NB classifier depends on the distribution of the class feature density. The accuracy is higher if the distribution is Gaussian although the number of training instance, s is small. On the other hand, the accuracy of NB is deteriorated if the distribution of the class feature density is non-Gaussian [9, 24]. In terms of the impact of the number of instance, the histogram is closely similar to the class feature density when the number of instance is large, $s \to \infty$. In cases when the number of instance is small, only partial of the histogram is required to be similar to the class feature density.



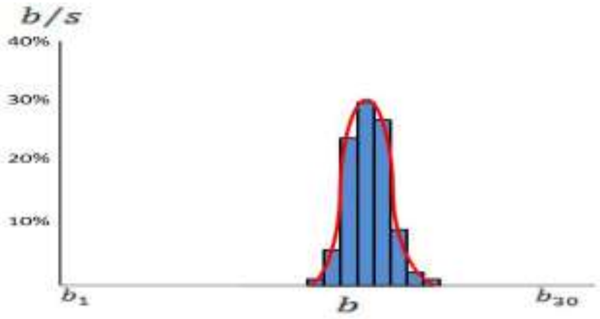
(a)

(b)

(c) (2)

(d)

(e)



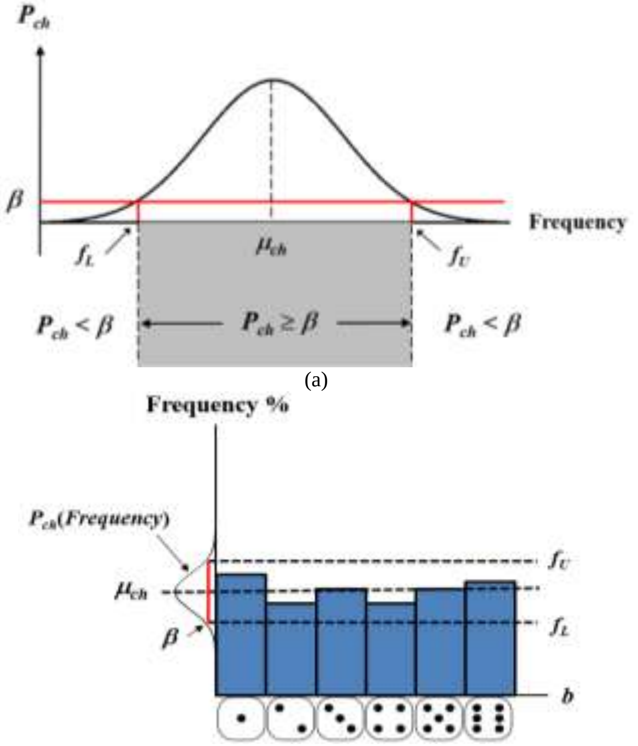
(f)

Fig. 1: Examples of some observed histograms derived from two different pdf(x). (a) and (d) are two different pdf(x). (b) and (e) are two observed histograms when the number of instance is small. (c) and (f) are two observed histograms when the number of instance is very large.

Fig. 1 illustrates the concept of using the observed histogram as a means to estimate the probability density. Here, two cases are presented using two different probability density functions [27] as shown in Fig. 1a and 1d. Fig. 1b and 1e show the observed histograms of the two density functions when the number of instance is small. Fig. 1c and 1f show the histograms of the two density functions when the number of instance is large.

The proposed supervised partial histogram Bayes (PHBayes), does not require the training of all the instance, but it is sufficient to rely on the instance that have high probability of change. In the PHBayes, the observed histogram is used to estimate the probability density and it is represented as **b** = (b$_1$,…, b$_r$), where r is the number of bin or interval. Each bin has probability of success, P = 1/r and probability of failure, Q = 1-P. In order to estimate the probability of change of one particular bin, b$_i$, De-Moivre-Laplace theorem [34], which treats the number of instance as trial, is applied. The probability of change, p$_{ch}$ is given in (4) where μ$_{ch}$ is the average level of noise or the level of chance in b$_i$ and ∑$_{ch}$ is the variance.

$$p_{ch}(b_i) \approx N(b_i, \mu_{ch}, \Sigma_{ch}) \qquad (4)$$



(a)

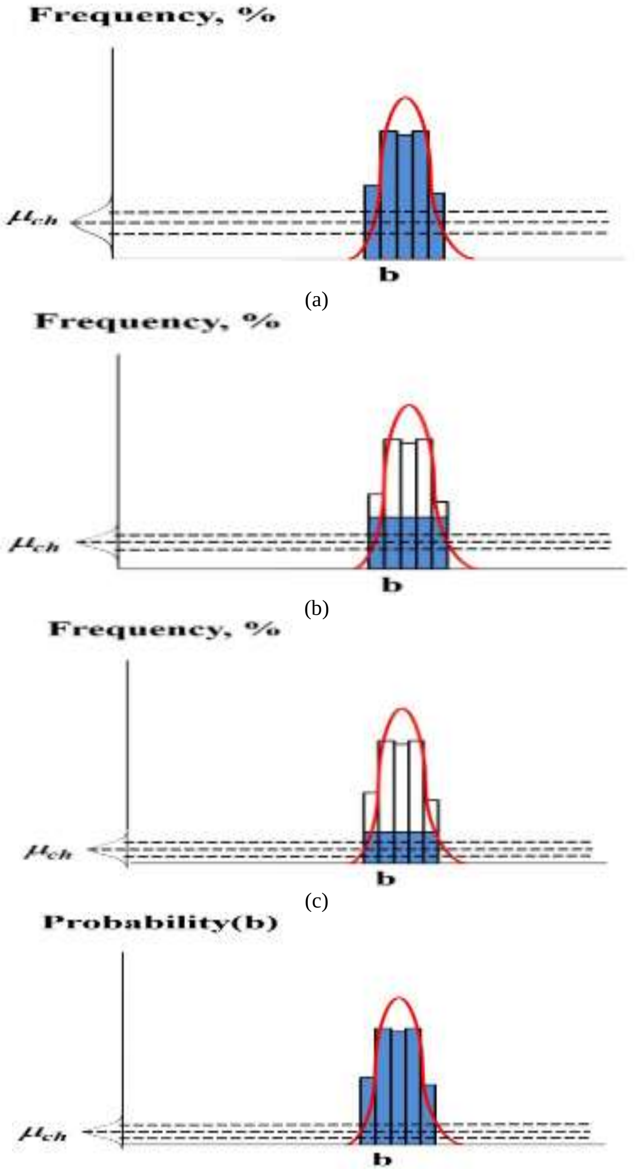
(b)

Fig. 2: (a) The probability of chance. (b) An example of the observed histogram after several trials on a six-sided dice. Probability of change estimation, p$_{ch}$ overlaying the observed histogram.

The probability of change, p$_{ch}$ is illustrated in Fig. 2a. With the increment of the number of trial, s, the distribution of the p$_{ch}$ is closely resembling Gaussian distribution. A horizontal line, β is used to control the noise level in the histogram by setting its level at the p$_{ch}$ axis. The value of β is 0 ≤ β ≤ 1. The intersection points between the p$_{ch}$ curve and the β line determine the lower frequency, f$_L$ and upper frequency, f$_U$ of the p$_{ch}$. A gray region between the f$_L$ and f$_U$ occurred when $p_{ch} \geq \beta$. Then, the probability of change, p$_{ch}$ is applied to the histogram as shown in Fig. 2b. The horizontal dashed lines indicate the level of noise, μ$_{ch}$, the lower frequency, f$_L$, and the upper frequency, f$_U$. If the p$_{ch}$ (b$_k$) < β , then the instance of b$_i$ with frequency higher than f$_U$ are subtracted from s resulting in the remaining instance, s$_{ch}$ as provided in (5) and the new values for the μ$_{ch}$ and ∑$_{ch}$ are updated. The upper frequency, f$_U$ is determined in (6).

$$s_{ch} = s - \sum_{i=1}^{r}(b_i - f_U \mid p_{ch}(b_i) < \beta \,\&\, b_i > \mu_{ch}) \qquad (5)$$

$$f_U = \mu_{ch} + \sqrt{-2\Sigma_{ch} \times \ln(\beta - \frac{1}{\sqrt{2\pi\Sigma_{ch}}})} \qquad (6)$$



(a)

(b)

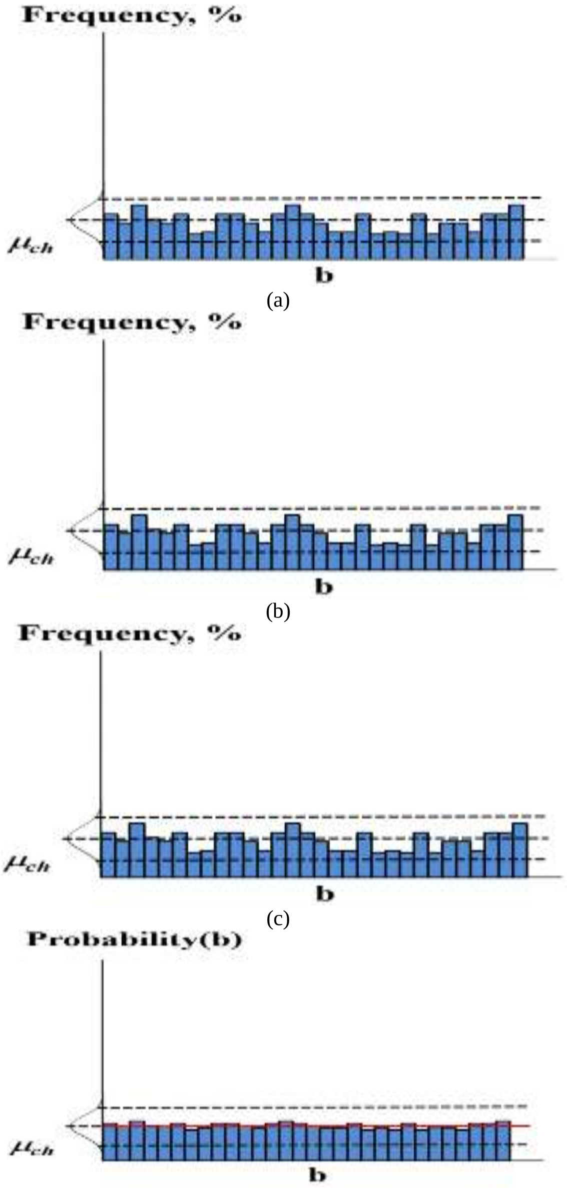
(c)

(d)

Fig. 3: Probability density estimation from b when p$_{ch}$(b$_i$) < β. (a)(b)(c) Several iteration stages of the probability estimation. (d) The probability density.

PHBayes estimates the probability distribution of class density p(b) based on the distribution of the histogram itself. There are two conditions that determine the changes of the histogram. The first condition is when $p_{ch}(b_i) < \beta$, then the probability distribution $p(b_i)$ remains unchanged. This situation can be illustrated in an example in Fig. 3. Here, the observed histogram can be derived from limited number of training instance, s. Here, the μ_{ch} and $\sum_{ch}$ are calculated from the bins of the histogram and the probability of change, p_{ch} is applied to the histogram, which indicates that $p_{ch}(b_i) < \beta$ and the frequency is higher than f_U as shown in Fig. 3a. The instance of b_i that are higher than the upper frequency, f_U are subtracted from the number of instance, s, resulting in the remaining instance, s_{ch}. The μ_{ch} and $\sum_{ch}$ are updated as shown in Fig. 3b. This process is repeated and the p_{ch}, s_{ch}, μ_{ch} and $\sum_{ch}$ are updated. The iteration stops when μ_{ch} and $\sum_{ch}$ converge. Then, the updated probability of change, p_{ch} is compared with the observed histogram. Since the peaks of the $p_{ch}(b_i)$ lie in the region of $p_{ch}(b_i) < \beta$, the probability distribution, p(b) remains unchanged as shown in Fig. 3d. In the case of $p_{ch}(b_i) < \beta$, probability distribution of class density $p(b_i)$ is based on the original histogram that is rescaled with the number of instance, s, as in (7).

$$p(b_i) = \frac{b_i}{s} \tag{7}$$

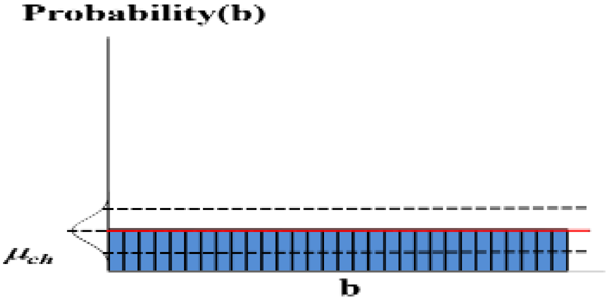


(a)

(b)

(c)

(d)



(e)

Fig. 4: Probability density estimation from b when $p_{ch}(b_i) > \beta$. (a)(b)(c) Several iteration stages of the probability estimation. (d)(e) The probability density.

The second condition is when $p_{ch}(b_i) \geq \beta$ and the frequency is lower than f_U , then there will be change at $p(b_i)$. The μ_{ch} and $\sum_{ch}$ of the probability of change, p_{ch} are not converge even after maximum iteration, as shown in Fig. 4a-c. Similar to the first case, the probability of change, p_{ch} is compared with the observed histogram. Since the peaks of the $p_{ch}(b_i)$ lie in the region of $p_{ch}(b_i) \geq \beta$, the $p(b_i)$ is rescaled close to the level of μ_{ch} in order to attenuate the noise of the histogram. The difference between the b_i and the μ_{ch} is multiplied with a weight of the attenuation, α and then shift with μ_{ch}. The value of α ranges from 0 to 1. Then, each bin of the histogram, b_i is rescaled with the number of instance, s and $p(b_i)$ for $p_{ch}(b_i) \geq \beta$ case is given in (8).

$$p(b_i) = \frac{\alpha(b_i - \mu_{ch}) + \mu_{ch}}{s} \tag{8}$$

For the two cases, the combination of $p(\mathbf{b}) = (p(b_1), \ldots, p(b_r))$ are used in the Bayesian rule for classification tasks. PHBayes assigns an instance to a class that has the highest posterior probability as given in (9).

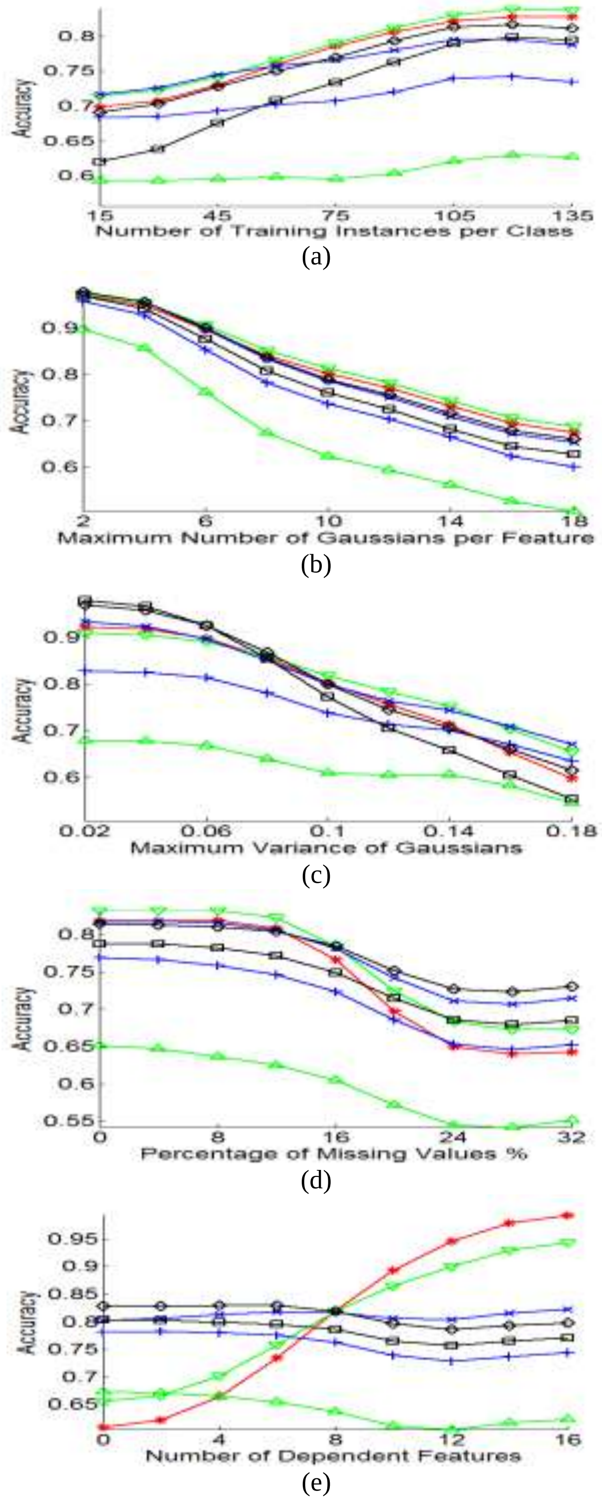
$$p(C \mid \mathbf{X})_{PHBayes} = p(C)\prod_{i=1}^{n} p(b_i) \tag{9}$$

4. Results and Discussion

The accuracy of classifier depends on several factors such as number of training instance, number of Gaussian, variance of the Gaussian, missing input values, and feature dependency. The default values for these factors are presented in Table 1. The analyses of impact of these factors were performed using synthetic data on the different classifiers: 1stNN, NCM, NSC, GMMC, PHbayes with r = 15, PHBayes(15), and PHBayes with r = 30, PHBayes(30). The value for the attenuation weight, α and the threshold, β of the PHBayes are set to α = 0.5 and β = 0.005. The generation of the synthetic data was based on the representation of pdfs of all features of the classes as mixture of Gaussians. For all the classes, instances were derived from the pdfs and the number of training instance was the same. For each class, 16 features (8 independents and 8 dependents) were used. Results are presented in Fig. 5.

Table 1: Default values of feature factors.

No.	Feature Factor	Default Value
1	Number of training instance	75 instances
2	Number of Gaussian	1 to 10 Gaussians
3	Variance of each Gaussian	2 to 10
4	Features dependency	8 independents and 8 dependents
5	Number of missing values	No missing values



(a)

(b)

(c)

(d)

(e)

Fig. 5: Several factors affecting the classification accuracy. (a) Number of instance. (b) Number of Gaussians. (c) Variance of Gaussians. (d) Missing values. (e) Dependent features.

Fig. 5a shows an analysis on the impact of the number of training instance as one of the factors that affects the classification accuracy. Number of training instance was varied from 15 to 135. The value for other factors were fixed at their respective default values. In general, the accuracy of all the classifiers increased with the addition of number of instance. However, the performance rate of the classifiers was at the different rate. The accuracy of the NB and NCM was less affected with the addition of the number of instance. PHBayes produced significant improvement as the number of instance increased. Large number of instance helped to improve the estimation of the histogram in the PHBayes. Fig. 5b shows the results of varying number of Gaussian that was ranges from 2 to 18. Other factors were set to their default values. As the number of mixture of Gaussian increased, the accuracy of all the classifiers deteriorated, especially NCM because it was less robust to the complexity of the data distribution. Fig. 5c shows the results of variance of Gaussian that ranges from 0.02 to 0.18. Other factors were set to their default values. The increment of the variance typically reduced the classification accuracy of all the techniques. Compared to the other classifiers, the declining rate of the accuracy of the PHBayes was more apparent because the class probability density was spread wider. The limitation of the PHBayes can be improved if large number of instance is used. Fig. 5d shows the results of missing input value in the instances. The percentage of the missing value ranges from 0 – 30%. Other factors were set to their default values. In general, the accuracy reduced when large number of input in the instance was missing. The declining trend of all the classifiers was highly correlated, but with different level of accuracy. PHBayes(15) was slightly better than other classifiers in handing missing input values because those values can be replaced with means of the PHBayes histogram. Fig. 5e shows the results of classifier accuracy with feature dependency. In this work, there were 16 features. For an example, if the number of dependent feature is set to 4, the remaining 12 features were independent. Here, the number of feature dependency varied from 0 to 16. Bayesian based classifiers like NB, GMMC, and PHBayes were not significantly affected with the varying number of feature dependency because all the features were assumed to be independents. In contrast, distance based classifier like 1stNN and NSC show significant improvement in their accuracy when the number of feature dependency increased.

Twenty different databases from UCI machine learning database [35] containing more than 50,000 instances from different classes were used. The database is presented in Table 2. Each database contains different classes, features, and instances. In order to standardize the analysis for all the classifiers, the features were scaled equally and nominal feature were ignored. About half of the instances were randomly selected as training instances and the other half were used for testing. PHBayes was compared with other classifiers, 1stNN, NSC, NCM, NB and GMMC.

Table 2: Twenty UCI databases used in the analyses.

No.	Database Name	No. of Instance	No. of Features	No. of Classes
1	Abalone	4177	8	3
2	Balance-scale	625	4	3
3	Blood Transfusion Service Center	748	4	2
4	Car evaluation	1728	6	4
5	Connectionist Bench (Sonar,	208	60	2
6	Contraceptive Method Choice	1473	9	3
7	Glass identification	214	9	6
8	Haberman's survival	306	3	2
9	Hayes-Roth	132	5	3
10	Hepatitis	155	19	2
11	Ionosphere	351	34	2
12	Iris	150	4	3
13	MAGIC Gamma Telescope	19020	10	2
14	Nursery	12960	8	5
15	Pima Indians Diabetes	768	8	2
16	Postoperative Patient	90	8	3
17	Spambase	4601	57	2
18	Statlog (Image Segmentation)	2310	19	7
19	Wine	178	13	3
20	Wisc. Breast Cancer (Diagnostic)	699	10	2

Table 3 presents the accuracy results of the classifiers on the databases. In general, PHBayes produced high accurate results in 12 of the 20 databases (60%) as compared to other classifiers. PHBayes(15) was the most accurate classifier with average accuracy 0.7930 and the most consistent technique that produced the most accurate classification for half of the databases. Table 4 shows results of the computational time for the training and testing stages for different classifiers. The analysis was performed using the 20 databases. Compare to other classifiers, PHBayes classifier demonstrated faster computational time in the training and testing stages.

Table 3: Accuracy of classification on the 20 UCI databases.

Data	1stNN	NCM	NSC	NB	GMMC	PHBayes (15)	PHBayes (30)
1	0.4745	0.4069	0.4314	0.4524	0.3558	0.4977	**0.5033**
2	0.8090	0.7500	0.6946	0.8974	**0.8978**	0.8724	0.8724
3	0.7067	0.6540	0.5971	**0.7610**	0.7254	0.7527	0.7447
4	**0.9043**	0.6965	0.7196	0.6884	0.7600	0.8167	0.8167
5	0.9240	**0.9962**	0.9702	0.9885	0.9865	0.9788	0.9904
6	0.4380	0.4363	0.4088	0.4798	0.3894	**0.5086**	0.5029
7	0.6402	0.5841	0.5654	0.5850	0.7280	**0.8533**	0.7523
8	0.7281	0.7523	0.6327	0.7556	**0.7595**	0.7536	0.7444
9	0.5444	0.5040	0.5207	0.6096	0.6126	**0.6409**	0.5929
10	0.6481	0.7104	0.6519	0.7182	0.6675	**0.8117**	0.8039
11	0.8029	0.7629	0.7651	0.7526	0.7766	**0.8474**	0.8343
12	0.9107	0.4520	0.8747	0.5053	0.8680	0.7720	**0.9173**
13	**0.9938**	0.9072	0.9243	0.9105	0.9728	0.9534	0.9910
14	**0.8903**	0.8205	0.7889	0.7165	0.7948	0.8703	0.8703
15	0.6464	0.6568	0.6362	0.6958	**0.6979**	0.6893	0.6771
16	0.5859	0.4504	0.5185	0.6015	0.3385	**0.6519**	**0.6519**
17	0.9143	0.8250	0.8393	0.7114	0.7956	**0.9619**	0.9615
18	0.4435	0.4386	0.4229	0.3612	0.6047	**0.7034**	0.7011
19	0.7270	0.7517	0.7629	0.8124	0.8315	**0.9483**	0.9247
20	0.9662	0.9731	0.9590	0.9668	0.9665	**0.9759**	**0.9759**
Average accuracy	0.7349	0.6764	0.6842	0.6985	0.7265	**0.7930**	0.7915

Table 4: Training and testing times on the 20 databases.

Algorithm	Training Time (s)	Testing Time (s)
1stNN	---	354.3157
NCM	1.7284	0.2485
NSC	22.2335	1.1067
NB	3.8252	0.5948
GMMC	202.8376	3.5611
PHBayes(15)	**1.5672**	0.1299
PHBayes(30)	1.8040	**0.1277**

The instances in the 1stNN need not be trained. Therefore, there was no training time. During the testing stage, 1stNN needed to calculate the distance between a testing instance and all the training instance in the databases. This factor made it as the slowest classifier during the testing stage. NCM calculated the class mean or the centroid from the training instances in the training stage. During the testing stage of the NCM, only the distance between a training instance and the class mean was measured. This factor helped to speed up the computational time of the NCM. The computational time of the NSC during the training and testing stages was slower than that of the NCM because NSC required clustering technique like k-means to estimate the subclass means for each class. Since k-means involved iteration process, NCS was relatively slower than the NCM. Moreover, NCM only used single class mean but NCS calculated a number of subclass means. The computational times for the training and testing stages of the NB were about two times slower than the NCM because it needed to calculate more parameters such as class mean, class variance, and assigning a testing instance to a class with the highest posterior probability. The training time for the GMMC was the slowest among all the classifiers because it represented class feature as mixture of Gaussian. Estimation of the mixture of Gaussian was performed using EM technique, which required few iterations and can slow down the training process. In the testing stage, no iterative procedure was involved, speeding up the computational time of the GMMC. The training and testing of the PHBayes were the fastest among other classifiers. During the training stage, the PHBayes was required to estimate the probability distribution derived from the observed histogram and to regenerate the histogram. The estimation of the probability density distribution involved iteration calculation of the probability of change. Unlike NSC and GMMC, PHBayes did not consider the number of training instance because the number of bin in the histogram was kept fixed and smaller than the number of training instance. During the testing stage, PHBayes only considered the posterior probability of the nearest bin to a testing instance, and not on the number of training instance.

5. Conclusion

This paper presents analyses on PHBayes and the comparison between other classifiers in the aspects of accuracy and computational time. Factors that affecting the classifier accuracy were investigated using synthetic data. These factors were number of training instance per class, number of Gaussian per feature, variance of Gaussian, percentage of missing value, and number of dependent feature. PHBayes was also tested on the real data. Results show that the PHBayes produced better accuracy in most of the cases. Increasing the number of instance improved the accuracy of the PHBayes. However, the number of instance did not required to be keep in the PHBayes memory because it relied on the number of bins in the observed histogram. In the training stage of the PHBayes, the number of instance was not considered in the class density calculation but it considered the number of bin in the observed histogram. This approach improved the computational time of the PHBayes, especially when the number of instance is large. As a result, PHBayes was faster than NSC, GMMC, and even NB. In the testing stage, the speed of the PHBayes was comparable with the NCM. Therefore, PHBayes is suitable for applications involving large number of instance because the accuracy of the PHBayes can be improved without consuming large space in its memory and not burdening the computational time in the training and testing stages. Future works will investigate the impact of the number of bin in the histogram, r towards the performance of the PHBayes.

Acknowledgement

Funding by the Ministry of Higher Education Malaysia under the research grant DIP-2015-012 is gratefully acknowledged.

References

[1] Freedman, S. T. and Adams, J. A. Filtering data based on human-inspired forgetting. IEEE Trans. Syst. Man Cybern. B, Cybern. 2011, 41(6):1544-1555.

[2] Mohammad, M. M., Jing, G., Latifur, K., Jiawei, H. and Bhavani, T. Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Trans. Knowl. Data Eng. 2011, 23(6):859-874.

[3] Bogdan, W. M. Neural network architectures and learning algorithms. IEEE Ind. Electron. Mag. 2009, 3(4):56-63.

[4] French, R.M. Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. 1999, 3(4):128-135.

[5] Mensink, T., Verbeek, J., Perronnin, F. and Csurka, G. Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35(11):2624-2637.

[6] Mathur, A. and Foody, G. M. Multiclass and binary SVM classification: Implications for training and classification users. IEEE Geosci. Remote Sens. Lett. 2008, 5(2):241-245.

[7] Chih-Wei, H. and Chih-Jen, L. A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. 2002, 13(2):415-425.

[8] Ravikumar, B., Thukaram, D. and Khincha, H. P. Comparison of multiclass SVM classification methods to use in a supportive system for distance relay coordination. IEEE Trans. Power Del. 2010, 25(3):1296-1305.

[9] Chaudhuri, P., Ghosh, A. K. and Oja, H. Classification based on hybridization of parametric and nonparametric classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 2009, 31(7):1153-1164.

[10] [10] Veenman, C. J. and Reinders, M. J. T. The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27(9):1417-1429.

[11] Viswanath, P. and Sarma, T. H. An improvement to k-nearest neighbor classifier. Proceedings of the Recent Advances in Intelligent Computational Syst., pp. 227-231, 2011.

[12] Yiguang, L., Sam, G. S., Chunguang, L. and Zhisheng, Y. k-NS: A classifier by the distance to the nearest subspace. IEEE Trans. Neural Netw. 2011, 22(8):1256-1268.

[13] Hui, W. Neighborhood counting measure and minimum risk metric. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32(4):766-768.

[14] Xueyi, W. A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality. Proceedings of the Int. Joint Conf. on Neural Networks, pp. 1293-1299, 2011.

[15] Hui, W. Nearest neighbors by neighborhood counting. IEEE Trans. Pattern Anal. Mach. Intell. 2006, 28(6):942-953.

[16] Yoonho, H., Bohyung, H. and Hee-Kap, A. A fast nearest neighbor search algorithm by nonlinear embedding. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3053-3060, Providence, 2012.

[17] Moutafis, P., Leng, M., and Kakadiaris, I. A. An overview and empirical comparison of distance metric learning methods. IEEE Trans. Cybern. 2017, 47(3):612-625.

[18] Yu, J., Bai, M., Wang, G. and Shi, X. Fault diagnosis of planetary gearbox with incomplete information using assignment reduction and flexible naive Bayesian classifier. J. of Mechanical Science and Technology, 2018, 32(1):37-47.

[19] Yonghong, H., Englehart, K. B., Hudgins, B. and Chan, A. D. C. A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. IEEE Trans. Biomed. Eng. 2005, 52(11):1801-1811.

[20] Karthikeyan, G., Rajendra, A. U., Chua Kuang, C., Choo Min L. and Thomas, A. K. One-class classification of mammograms using trace transform functionals. IEEE Trans. Instrum. Meas. 2014, 63(2):304-311.

[21] Wei, L., Saurabh, P., James, F. E. and Mann, B. L. Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. IEEE Trans. Geosci. Remote Sens. 2012, 50(4):1185-1198.

[22] Saurabh, P., Minshan, C., Wei, L. and James, F. E. Segmented mixture-of-Gaussian classification for hyperspectral image analysis. IEEE Geosci. Remote Sens. Lett. 2014, 11(1):138-142.

[23] Wei, L., Saurabh, P. and James, F. E. Hyperspectral image classification using Gaussian mixture models and Markov random fields. IEEE Geosci. Remote Sens. Lett. 2014, 11(1):153-157.

[24] Xi-Zhao, W., Yu-Lin, H. and Wang, D. D. Non-naive Bayesian classifiers for classification problems with continuous attributes. IEEE Trans. Cybern. 2014, 44(1):21-39.

[25] John, G. H. and Langley, P. Estimating continuous distributions in Bayesian classifiers. Proceedings of the Conf. on Uncertain. Artif. Intell. pp. 338–345, Montréal, 1995.

[26] Liu, J. N. K., Yu-Lin, H., Xi-Zhao, W. and Yan-Xing, H. A comparative study among different kernel functions in flexible naive Bayesian classification. Proceedings of the Int. Conf. on Mach. Learning and Cybern., pp. 638-643, 2011.

[27] Lawend, H. O.; and Muad, A. M. A non-parametric partial histogram Bayes learning algorithm for classification applications. Proceedings of the IEEE Int. Conf. on Control Sys., Computing and Eng., pp. 35-39, 2014.

[28] Hung-Ju, H. and Chun-Nan, H. Bayesian classification for data from the same unknown class. IEEE Trans. Syst. Man Cybern. B, Cybern. 2002, 32(2):137-145.

[29] Soria, D., Garibaldi, J. M., Biganzoli, E. and Ellis, I. O. A comparison of three different methods for classification of breast cancer data. Proceedings of the Seventh Int. Conf. on Mach. Learning and Appl , pp. 619-624, 2008.

[30] Yan-Shi, D. and Ke-Song, H. A comparison of several ensemble methods for text categorization. Proceedings of the Services Computing, pp. 419-422, 2004.

[31] Dilrukshi, I. and Zoysa, K. D. Twitter news classification: Theoretical and practical comparison of SVM against naive Bayes algorithms. Proceedings of the Int. Conf. on Advances in ICT for Emerging Regions, pp. 278-278, 2013.

[32] Shameem, F. and Nisar, H. Comparison of classification techniques- Proceedings of the SVM and Naives Bayes to predict the arboviral disease-dengue. IEEE Int. Conf. on Bioinformatics and Biomedicine Workshops, pp. 538-539. 2011.

[33] Perez, A., Larranaga, P. and Inza, I. Bayesian classifiers based on kernel density estimation: Flexible classifiers. Int. J. of Approximate Reasoning. 2009, 50(2):341-362.

[34] Papoulis, A. Probability, random variables, and stochastic processes. McGraw-Hill. 1991.

[35] Asuncion, A. and Newman, D.J. UCI machine learning repository. 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.