Improving Performance of Classifiers using Rotational Feature Selection Scheme

Shib Sankar Bhowmick[1, 2], Indrajit Saha[1]
[1]Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700032,West Bengal,
India.
[2]Department of Informatics, University of Evora,
Evora 7004-516, Portugal.
e-mail: shibsankar.ece@gmail.com, indra@icm.edu.pl

Luis Rato[2], Debotosh Bhattacharjee[1]
[1]Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700032,West Bengal,
India.
[2]Department of Informatics, University of Evora,
Evora 7004-516, Portugal.
e-mail: lmr@di.uevora.pt, debotosh@ieee.org

***Abstract*—The crucial points in machine learning research are that how to develop new classification methods with strong mathematic background and/or to improve the performance of existing methods. Over the past few decades, researches have been working on these issues. Here, we emphasis the second point by improving the performance of well-known supervised classifiers like Naive Bayesian, Decision Tree and k-Nearest Neighbor. For this purpose, recently developed rotational feature selection scheme is used before performing the classification task. It splits the training data set into different number of rotational non-overlapping subsets. Subsequently, principal component analysis is used for each subset and all the principal components are retained to create an informative set that preserve the diversity of the original training data. Thereafter, such informative set is used to train and test the classifiers. Finally, posterior probability is computed to get the classification results. The effectiveness of the rotational feature selection integrated classifiers is demonstrated quantitatively by comparing with aforementioned classifiers for 10 real-life data sets. Finally, statistical test has been conducted to show the superiority of the results.**

Keywords: Decision Tree; k-NN; Naive Bayesian; Principal Component Analysis; Rotational Feature Selection; Statistical Test.

I. INTRODUCTION

Classification is an important problem in data mining research [1-5]. It has been studied extensively by the mathematicians and computer science engineers to find a possible solution for knowledge acquisition or extraction. One of the main issues in classification task is to improve the efficiency of the existing classifiers. During the last decades, considerable attentions have been noticed for this task.

In this study we have used, Naive Bayesian (NB) [6], Decision Tree (DT) [5] and k-Nearest Neighbor (k-NN) [7] classifiers to improve the performance. In this regards, rotational feature selection (RFS) [8] scheme is used to generate an informative set that can be used during testing and training for these classifiers. The RFS scheme works with Principal Component Analysis (PCA) to preserve the variability information of the rotational non-overlapping subsets of original data. Here the main motivation is to make a better diversified and accurate classifier. Diversity is achieved by using PCA, where it uses to extract the principal components of rotational subsets for the classifier and preservation of all principal components increases the accuracy.

The experimental studies conducted with available 10 real-life data sets from UCI repository [9]. The results show that all these classifiers, with rotational feature selection scheme can produce significantly higher accuracy more often than the conventional Naive Bayesian, Decision Tree and k-NN classifiers. Subsequently, *t*-test [10], confusion matrix [11] and *Kappa* index [12] have been used to establish the superiority of the results produced by classifiers in conjunction with RFS scheme.

The rest of this paper is organized as follows: Section 2 briefly describes the Naive Bayesian, Decision Tree and k-NN classifiers. The proposed RFS integrated classification scheme is discussed in Section 3. Experimental study has been conducted in Section 4. Finally, Section 5 concludes this paper with an additional note of future work.

II. BRIEF DESCRIPTION OF CLASSIFIERS

A. Naive Bayesian Classifier

The Naive Bayes (NB) classifier [13], [14] is developed based on the Bayes' theorem. It assumes that the attributes or features are conditionally independent for the given class label to compute the class-conditional probability. Therefore, the assumption of conditional independence is defined as follows:

$$P(X|\ =\ ) = \prod_{i=1}^{d} P(X_i\ |\ =\ ) \tag{1}$$

where each attribute set $X = X_1, X_2, \ldots, X_d$ consists of d attributes. Thereafter, it uses to compute the conditional probability of each X_i for given . In order to classify a test data, the classifier computes the posterior probability for each class and it is defined as follows:

$$P(|X) = \frac{P()\prod_{i=1}^{d} P(X_i|\ )}{P(X)} \tag{2}$$

Here, the posterior probabilities are computed by multiplying the prior probabilities with the class-conditional

probabilities. The prior probability of each class is calculated by the fraction of training points that belong to each class.

B. Decision Tree Classifier

C4.5 [5] is a decision tree generating algorithm and the extended version of ID3 algorithm. Both the algorithms are developed by Ross Quinlan. Moreover, the decision trees generated by C4.5 are often used for classification, thus it is also known as statistical classifier. To classify the data points, C4.5 uses the concept of entropy to build the decision trees from a set of training data. For this purpose, at every step, the highest information gain attribute is considered. Based on that attribute, decision is taken to split the training set into one or two subsets. The process will continue recursively until all nodes are exhausted. Thereafter, depending on user given parameters, C4.5 prunes the generated tree in order classifies the test data points.

C. k-Nearest Neighbor Classifier

k-Nearest Neighbor (k-NN) [7] classifier is one of the earliest, simple and popular classifier. The algorithm known as k-NN decision rule, can be stated as follow. Consider an unknown pattern vector *x* and a distance measure,

- At first out of *N* training patterns, *k* nearest neighbors, irrespective of the class labels, are chosen.
- Thereafter, the unknown vector *x* is assigned to the class φ_i , *i =1, 2,...,c* for which $\sum_i k_i$ is maximum, where k_i, denotes the neighboring pattern belonging to class $_i$.

For $k = 1$ the algorithm is simply known as nearest neighbor rule. Various distance measures can be used including the simplest Euclidean and Mahalanobis. For large values of *N*, this simple classifier can show quite a good performance.

III. PROPOSED RFS INTEGRATED CLASSIFICATON SCHEME

Consider a training set $£ = \{(x_i, _i)\}_{i=1}^{N}$ consisting of *N* independent data points, in which each $(x_i, _i)$ is described by an input attribute vector $x_i = (x_{i1}, x_{i2,\ldots,} x_{id}) \epsilon^{d}$ and a class label $_i$. $_i$ takes a value from the label space $\{\varphi_1, \varphi_2, \ldots, \varphi_c\}$. The main objective of a classification task is to use the information only from £ to construct a classifier which performs well on unseen data. For simplicity of the notations, let *X* be a $N \times d$ data matrix composed with the values of *d* input attributes for each training instance and be a column vector of size *N*, containing the outputs of each training instance in £. Moreover, £ can also be expressed by concatenating *X* and vertically, that is $£ = [X]$. Also let $F = \{X_1, X_2,...,X_d\}^T$ be the attribute or feature set composed of *d* input attributes or features and *C* be the classifier.

Details of rotational feature selection method are mentioned in Algorithm 1, where to construct the training set for classifier, the feature set *F* is randomly split into *S*, subsets, known as F_s. The subsets are disjoint in nature, to maximize the chance for high diversity. Then a submatrix X_s is computed which corresponds to the attribute in F_s and from this submatrix a new bootstrap sample $X_s^{'}$ of size 75% of the samples are selected. Thereafter, PCA technique is applied to each subset to obtain a matrix D_s and the diversity of the data is preserved by retaining all the principal components. Thus, *S* axis rotations take place to form an informative attributes for a base classifier. Subsequently, the matrix D_s is arranged into a block diagonal matrix *R*. The training set for classifier *C* (which is Naive Bayesian or Decision Tree (C4.5) or k-NN classifier at a time) is constructed by rearranging the rows of *R*, so that they correspond to the original attributes in *F*. The rearranged rotation matrix is R^a and training set for classifier *C* is $[XR^a,]$.

In the testing phase, given a test sample *I*, let $C_j(IR^a)$ be the posterior probability produced by the classifier *C* on the hypothesis that *I* belongs to class φ_j. Then the confidence for a class is determined by the posterior probability. Formally, it can be defined as follows:

$$\Psi_j(I) = C_j(IR^a) \qquad (j = 1,2, \ldots, c) \qquad (3)$$

Thereafter, *I* is assigned to the class with the largest confidence. Note that while running the RFS algorithm to solve a classification task, parameter like *S* is needed to be specified in advance.

IV. EXPERIMENTAL RESULTS

The RFS scheme is integrated with Naive Bayesian, Decision Tree (C4.5) and k-NN classifiers and their effectiveness is demonstrated by comparing with conventional classifiers for 10 real-life data sets.

A. Data Sets

Table I gives the information about data sets with different characteristics. The first column of that Table giving the name of different data sets, The second and third columns respectively gives the sample size and number of classes of each data set. The last column summarizes the information of total number of input attributes. During pre-processing of each dataset, missing value instances are deleted from data sets.

B. Predicted Output analysis

The Naive Bayesian, Decision Tree (C4.5) and k-NN classifiers as well as RFS integrated scheme are implemented by using the Matlab software of version 7.1.

Algorithm 1 Rotational Feature Selection Integrated Classification Scheme

Require: For Training
X, Data Set
, Class Label
S, Number of Feature Sets
Require: For Testing
I, A data object to classify
Ensure:
Class label of *I*
Prediction Accuracy of Classifier

1:Randomly split the attribute set *F* into *S* subsets, F_s where $(s = 1, 2,..., S)$.
2:**for** $(s = 1, 2,...,S)$ **do**
3: Create submatrix X_s using *X* and F_s.
4: Create a new bootstrap sample $X_s^{'}$ of size 75% form X_s.
5: Apply PCA on $X_s^{'}$ to obtain the coefficient matrix D_s.
6: **end for**
7: Arrange the matrices D_s $(s = 1, 2,..., S)$ into a block diagonal matrix *R*.
8: Construct the rotation matrix R^a by rearranging the rows of *R*.
9: Train the classifier *C* using $[XR^a\ ,\ ]$ as the training set.
10: Test the sample *I* using *C* and compute posterior probability to assign class label.
11: **return** Class label of *I* and Prediction Accuracy of Classifier.

TABLE I. SUMMERY OF THE DATA SETS.

Data Set	Number of data points	Classes	Number of attributes
Abalone	4177	3	7
BCW	691	2	9
Car	1748	4	6
Dermatology	366	6	34
Glass	214	6	10
Liver	345	2	6
Pima	768	2	8
Sonar	208	2	60
Vehicle	94	4	18
Yeast	1484	10	8

The value of *S* (*S* Number of feature sets) is not fixed for each data set, thus we adjusted it manually depending on the attribute numbers of the data sets. Here, each method executed 20 times and their prediction accuracy is summarized by computing mean, standard deviation and *kappa* index [12]. Finally, statistical test has been conducted to show the superiority of the results produced by integrated RFS classifiers. Note that the value of k for k-NN classifiers is set to 13 for all the data sets.

In Table II, the mean and Standard deviation of the predicted accuracy (expressed in %) for each data set are described, to quantitatively judge the performance of integrated RFS classifiers. The classification methods are used twice, first the classifier are used with the integration of RFS scheme and second, only the conventionals' are used. For each data set and classifier, the values following "±" are their respective standard deviations. As can be seen from the results, classifiers with RFS method produce consistent better results for most of the data sets, but one or two odd cases are also seen. For example, "Yeast" data set using all classifiers gives abnormally low accuracy rates for both integrated RFS and conventional classifiers. In order to see, statistical superiority of the integrated RFS classifiers a one-tailed paired *t*-test [10] is performed with the significance level $\alpha = 0.05$. The results for which a significant difference between conventional classifiers and integrated RFS classifiers are found and marked with a bullet or an open circle next to the values of standard deviation (SD) in Table II. A bullet beside any SD result denotes that integrated RFS classifiers are significantly better than conventional classifiers and an open circle next to any SD result shows that conventional classifiers are better than the integrated RFS classifiers.

In Table III, "Win-Tie-Loss" information is given where the significant difference in performance between the integrated RFS classifiers and the corresponding conventional classifiers are denoted by "Win" values. The "Tie" indicates the number of data sets on which the difference between the performance of RFS integrated classifiers and its corresponding algorithm is not significant,

TABLE II. MEAN AND STANDARD DEVIATION OF PREDICTION ACCURACY (EXPRESSED IN %) FOR 20 RUNS OF EACH CLSSIFIER ON REAL-LIFE DATA SETS.

	RFS Integrated Classifier						**Conventional Classifier**					
Data Set	***Naive Bayesian***		***Decision Tree***		***k-NN***		***Naive Bayesian***		***Decision Tree***		***k-NN***	
Abalone	86.34	±0.76	86.78	±0.82	73.91	±0.54	86.20	±0.70	83.03	±1.12•	79.57	±0.78◦
BCW	96.85	±2.63	96.87	±0.30	96.38	±0.34	96.53	±3.88	94.47	±0.71•	96.38	±0.33
Car	95.99	±1.74	96.09	±0.41	93.58	±0.27	65.40	±3.68•	94.65	±0.51•	84.78	±0.29•
Dermatology	96.86	±1.13	96.98	±0.79	87.16	±0.78	95.67	±1.89•	94.63	±0.81•	80.61	±0.68•
Glass	75.44	±0.24	75.63	±0.84	89.07	±0.75	74.75	±0.88•	75.51	±0.83	88.61	±1.68•
Liver	69.41	±0.81	70.96	±1.29	74.49	±1.07	61.74	±1.48•	64.15	±2.31•	70.72	±1.99•
Pima	74.26	±1.16	75.21	±0.76	78.39	±0.38	71.55	±0.90•	70.60	±1.23•	77.99	±1.14•
Sonar	82.02	±1.89	81.37	±2.23	71.15	±1.84	80.02	±2.14•	69.90	±2.36•	76.92	±2.24◦
Vehicle	76.32	±2.09	76.35	±0.89	57.45	±0.97	68.50	±1.06•	69.32	±1.25•	52.13	±1.00•
Yeast	58.73	±0.77	59.96	±0.68	64.49	±0.82	57.76	±0.92•	52.32	±1.22•	55.32	±0.93•

"•" indicates that RFS integrated classifiers are significantly better and "◦" denotes that conventional classifiers without RFS are significantly worse at the significance level $\alpha = 0.05$.

TABLE III. ONE-TAILED PAIRED T-TEST RESULTS OF CLASSIFIERS

***t*-test Result**	**RFS Integrated NB vs. Conventional NB**	**RFS Integrated DT vs. Conventional DT**	**RFS Integrated k-NN vs. Conventional k-NN**
Win	8	9	7
Tie	2	1	1
Loss	0	0	2

and the worst significant difference in performance between the integrated RFS classifiers and its conventional classifiers are denoted by "Loss" values. For example, when RFS integrated k-NN classifier compared with its conventional classifier, the statistically significant difference is favorable to RFS integrated scheme in 7 sets, unfavorable in 2 sets and not significant in 1 set. Similar results are observed for other RFS integrated classifiers.

Like statistical *t*-test, Confusion matrix [11] is also used to judge the performance of different classifiers. Each classified instance is mutually exclusively located in the confusion matrix. The diagonal cells in the matrix, gives information about the correctly classified instances where all the off diagonal cells represent miss classified instances. Here, confusion matrixes of all the data sets for all classifiers are computed. Among them, best four confusion matrices of Vehicle, Glass, Dermatology and Car data sets for RFS integrated Naive Bayesian classifier are shown in Figure 1. Along with Confusion matrix [11], *Kappa* index [12] is used here to justify the accuracy assessment of different classifiers. The higher value of *kappa* (close to one) indicates better accuracy. For most of the cases, it has been found from Table IV that the *kappa* values are better for RFS integrated classifiers than there corresponding conventional classifiers.

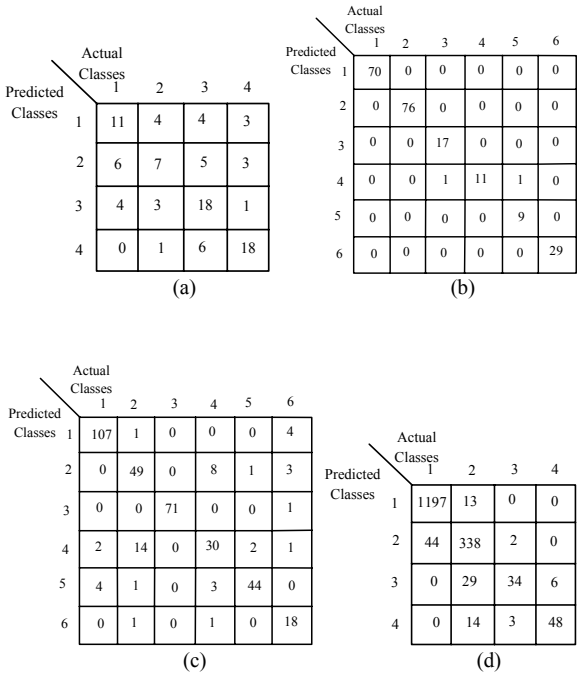

Figure 1. Best Confusion matrix out of 20 runs for (a) Vehicle and (b) Glass (c) Dermatology and (d) Car data sets.

TABLE IV. AVERAGE *KAPPA* INDEX FOR DIFFERENT DATA SETS.

Data Set	RFS Integrated Classifier			Conventional Classifier		
	Naive Bayesian	***Decision Tree***	***k-NN***	***Naive Bayesian***	***Decision Tree***	***k-NN***
Abalone	0.83	0.83	0.71	0.83	0.80	0.76
BCW	0.98	0.98	0.98	0.98	0.95	0.98
Car	0.97	0.97	0.94	0.63	0.95	0.81
Dermatology	0.98	0.98	0.84	0.96	0.95	0.77
Glass	0.72	0.72	0.86	0.71	0.72	0.85
Liver	0.67	0.68	0.71	0.59	0.62	0.68
Pima	0.71	0.72	0.75	0.69	0.67	0.75
Sonar	0.79	0.78	0.69	0.77	0.67	0.74
Vehicle	0.73	0.73	0.55	0.66	0.66	0.50
Yeast	0.55	0.57	0.61	0.54	0.50	0.52

V. CONCLUSIONS

This paper presents a novel performance enhancement technique for classifiers using rotational feature selection scheme. The performance of the classifiers like Naive Bayesian, Decision Tree (C4.5) and k-NN is boosted with the integration of RFS scheme. For this process, first the data set is divided into different subsets, thereafter principal component analysis separately run on each subset to generate an informative subsets, which are reassembled later by keeping all the principal components. As a result, the original data is transformed linearly into an informative set. Thereafter, such set is used for training and testing the classifiers. Finally, the classification is done by computing the posterior probability. The experimental results, demonstrate the effectiveness of the RFS integrated classifiers quantitatively by comparing it with conventional classification techniques for 10 real-life data sets. Statistical test like one-tailed paired *t*-test has been performed and that also suggests the superiority of the results produced by RFS integrated classifiers.

Our future research, will aim to improve the classifier performance further. In this regards, different feature selection algorithms such as ICA, sub PCA, ect. can be used. Moreover, RFS integrated classifiers can be applied for pixel classification of satellite imagery [15-18], microarray classification [19], protein translational modification site prediction [20, 21], human leukocyte antigen class II binding peptide prediction [22] ect. The authors are currently working towards achieving these goals.

ACKNOWLEDGMENT

This work is supported in part by Erasmus Mundus Mobility with Asia (EMMA) grant 2012 from the European Union at the Department of Informatics, University of Evora, Portugal and University with Potential for Excellence (UPE) - Phase II project grant from University Grants Commission (UGC) in India.

REFERENCES

[1] M. Kukar, "Quality assessment of individual classifications in machine learning and data mining," Knowledge Information System, vol. 9(3), pp. 364-384, 2006.

[2] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," Machine Learning, vol. 29, pp.103-130, 1997.

[3] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18(6), pp.607-616, 1996.

[4] L. Reyzin and R.E. Schapire, "How boosting the margin can also boost classifier complexity," in Proceedings of the 23rd international conference on machine learning, pp. 753-760, 2006.

[5] J. R. Quinlan, "C4.5: Programs for machine learning," Morgan Kaufmann Publishers, 1993.

[6] C. C. Hsua, Y. P. Huanga and K. W. Changa, "Extended Naive Bayes classifier for mixed data," Expert Systems with Applications, vol. 35(3), pp. 1080-1083, 2008.

[7] D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langerman, P. Morin and G. Toussaint, "Output-sensitive algorithms for computing nearest-neighbor decision boundaries," Discrete and Computational Geometry, vol. 33(4), pp. 593-604, 2005.

[8] J. J. Rodriguez, L. I. Kuncheva and C. J. Alonso, "Rotation Forest: A new classifier ensemble method," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28(10), pp. 1619-1630, 2006.

[9] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," www.ics.uci.edu/~mlearn/mlrepository. html, 1998.

[10] G. A. Ferguson and Y. Takane, "Statistical analysis in psychology and education," McGraw-Hill Publishers, 2005.

[11] N. Marom, L. Rokach and A. Shmilovici, "Using the confusion matrix for improving ensemble classifiers," in IEEE 26th convention of electrical and electronics engineers, pp. 555-559, 2010.

[12] J. A. Cohen, "Coefficient of agreement for nominal scales," Educational and Psychological Measurement, vol. 20(1), pp. 37-46, 1960.

[13] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis," John Wiley & Sons, 1973.

[14] P. Langley, W. Iba and K. Thompson, "An analysis of bayesian classifiers," in Proceedings of the international conference on artificial intelligence, pp. 223-228, 1992.

[15] I. Saha, U. Maulik and D. Plewczynski, "A new multi-objective technique for differential fuzzy clustering," Applied Soft Computing, vol. 11(2), pp. 2765-2776, 2011.

[16] I. Saha, U. Maulik, S. Bandyopadhyay and D. Plewczynski, "SVMeFC: SVM ensemble fuzzy clustering for satellite image segmentation," IEEE Geoscience and Remote Sensing Letters, vol.9(1), pp. 52-55, 2011.

[17] U. Maulik and I. Saha, "Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery," Pattern Recognition, Vol. 42(9), pp. 2135-2149, 2009.

[18] U. Maulik and I. Saha, "Automatic Fuzzy Clustering using Modified Differential Evolution for Image Classification," IEEE Transactions on Geoscience and Remote Sensing, Vol. 48(9), pp. 3503-3510, 2010.

[19] I. Saha, U. Maulik, S. Bandyopadhyay and D. Plewczynski, "Improvement of new automatic differential fuzzy clustering using SVM classifier for microarray analysis," Expert Systems with Applications, Vol. 38(12), pp. 15122-15133, 2011.

[20] D. Plewczynski, S. Basu and I. Saha, "AMS 4.0: Consensus Prediction of Post-Translational Modifications in Protein Sequences," Amino Acids, Vol. 43(2), pp. 573-582, 2012.

[21] I. Saha, U. Maulik, S. Bandyopadhyay and D. Plewczynski, "Fuzzy Clustering of Physicochemical and Biochemical Properties of Amino Acids," Amino Acids, Vol. 43(2), pp. 583-594, 2012.

[22] I. Saha, G. Mazzocco and D. Plewczynski, "Consensus classification of Human Leukocyte Antigens class II proteins," Immunogenetics, Vol. 65, pp. 97-105, 2013.