# RotaSVM: A New Ensemble Classifier

Shib Sankar Bhowmick[1,2,⋆], Indrajit Saha[1,⋆],
Luis Rato[2], and 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
shibsankar.ece@gmail.com, indra@icm.edu.pl,
lmr@di.uevora.pt, debotosh@ieee.org

**Abstract.** In this paper, an ensemble classifier, namely RotaSVM, is proposed that uses recently developed rotational feature selection approach and Support Vector Machine classifier cohesively. The RotaSVM generates the number of predefined outputs of Support Vector Machines. For each Support Vector Machine, the training data is generated by splitting the feature set randomly into $\mathcal{S}$ subsets. Subsequently, principal component analysis is used for each subset to create new feature sets and all the principal components are retained to preserve the variability information in the training data. Thereafter, such features are used to train a Support Vector Machine. During the testing phase of RotaSVM, first the rotation specific Support Vector Machines are used to test and then average posterior probability is computed to classify sample data. The effectiveness of the RotaSVM is demonstrated quantitatively by comparing it with other widely used ensemble based classifiers such as Bagging, AdaBoost, MultiBoost and Rotation Forest for 10 real-life data sets. Finally, a statistical test has been conducted to establish the superiority of the result produced by proposed RotaSVM.

**Keywords:** Principal component analysis, rotational feature selection, statistical test, support vector machine.

## 1 Introduction

Integration of classifiers nowadays is drawing much attention of the machine learning and pattern recognition communities and growing rapidly [1–10]. In integrated classification techniques, an ensemble of classifiers is generated by giving similar or different permutated training data sets. Thereafter, class label of the test sample is assigned by either majority voting or averaging the output probabilities of the ensemble. Recent research shows that ensemble based classifiers, such as Bagging [11], AdaBoost [12, 13], Random Forest [14] and Rotation

⋆ Both the authors contributed equally.

M. Emmerich et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Advances in Intelligent Systems and Computing 227,
DOI: 10.1007/978-3-319-01128-8_4, 

Forest [15], are used more often to increase the prediction accuracy of learning systems [9, 16–20].

Among these ensemble classifiers, Rotation Forest [15] performs much better than other ensemble methods. It uses rotational feature sets for decision tree classifiers. In Rotation Forest, rotational feature sets subsequently undergo Principal Component Analysis (PCA) to preserve the variability information of the training data. Here the main idea is to simultaneously increase diversity and individual accuracy within the decision tree classifiers. Diversity is achieved by using PCA, which is used to extract the principal components of rotational features for each classifier and accuracy is sought by keeping all principal components [15, 21].

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser *et al.* [22]. The basic idea of SVM is to find a hyperplane which separates the $d$-dimensional data perfectly into two classes. However, since classification data is often not linearly separable, SVM introduced the notion of a "kernel induced feature space" which embed the data into a higher dimensional feature space where the data is linearly separable. For this purpose, first the hyperplane is found, which separates the largest possible fraction of points such that points on the same side belong to the same class, while the distance of each class from the hyperplane is maximized.

As both Rotation forest and Support Vector Machine are successfully used in classification, thus their integration may achieve even higher prediction accuracy than either of them. Hence, in this paper, an ensemble classifier, named as RotaSVM, is proposed by integrating rotational feature selection scheme with SVM. The RotaSVM produces the number of predefine outputs of SVMs. For each SVM, the training data is generated by splitting the feature set randomly into $\mathcal{S}$ subsets. Subsequently, principal component analysis is used for each subset to create new feature sets and all the principal components are retained to preserve the variability information of the training data. Thereafter, such features are used to train a Support Vector Machine. During the testing phase of RotaSVM, the sample data are the input to the rotation specific Support Vector Machines. Subsequently, it is classified by computing average posterior probability. The experimental studies were conducted with available 10 real-life data sets[1]. The results show that RotaSVM can produce significantly lower prediction error more often than Rotation Forest and other ensemble based classifiers such as Bagging, AdaBoost and MultiBoost for all the data sets. Finally, $t$-test [24] has been conducted to establish the statistical significance of the result produced by RotaSVM.

The rest of this paper is organized as follows: Section 2 briefly describes the Support Vector Machine classifier. The proposed RotaSVM is discussed in Section 3. Section 4 shows the empirical results. Finally, Section 5 concludes this paper with an additional note of future work.

[1] UCI repository [23].

## 2 Brief Description of Support Vector Machine

In this section, we briefly discuss Support Vector Machine classifier which is inspired by statistical learning theory. It performs structural risk minimization on a nested set structure of separating hyperplanes [25]. Viewing the input data as two sets of vectors in a $d$-dimensional space, an SVM constructs a separating hyperplane in that space, one which maximizes the margin between the two classes of points. To compute the margin, two parallel hyperplanes are constructed on each side of the separating one. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes. The larger margins or distances between these parallel hyperplanes indicate better generalization error of the classifier. Fundamentally, the SVM classifier is designed for two-class problems. It can be extended to handle multiclass problems by designing a number of one-against-all or one-against-one twoclass SVMs [26]. For example, a $K$-class problem is handled with $K$ two-class SVMs [27]. For linearly nonseparable problems, SVM transforms the input data into a very high-dimensional feature space and then employs a linear hyperplane for classification. Introduction of a feature space creates a computationally intractable problem. SVM handles this by defining appropriate kernels so that the problem can be solved in the input space itself. The problem of maximizing the margin can be reduced to the solution of a convex quadratic optimization problem, which has a unique global minimum.

For a binary classification training data problem, suppose a data set consists of $N$ feature vectors $(x_i, y_i)$, where $y_i \in \{+1, -1\}$, denotes the class label for the data point $x_i$. The problem of finding the weight vector $\nu$ can be formulated as minimizing the following function:

$$L(\nu) = \frac{1}{2} \|\nu\|^2 \tag{1}$$

subject to

$$y_i[\nu \cdot \phi(x_i) + b] \geq 1, i = 1, \ldots, N \tag{2}$$

Here, $b$ is the bias and the function $\phi(x)$ maps the input vector to the feature vector. The SVM classifier for the case on linearly inseparable data is given by

$$f(x) = \sum_{i=1}^{N} y_i \beta_i K(x_i, x) + b \tag{3}$$

where $K$ is the kernel matrix, and $N$ is the number of input patterns having nonzero values of the Langrangian multipliers $\beta_i$. These $N$ input patterns are called support vectors, and hence the name SVMs. The Langrangian multipliers $\beta_i$ can be obtained by maximizing the following:

$$Q(\beta) = \sum_{i=1}^{N} \beta_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \beta_i \beta_j K(x_i, x_j) \tag{4}$$

subject to

$$\sum_{i=1}^{N} y_i \beta_i = 0 \qquad 0 \leq \beta_i \geq C, \quad i = 1, ...., N \tag{5}$$

where $C$ is the cost parameter, which controls the number of non separable points. Increasing $C$ will increase the number of support vectors thus allowing fewer errors, but making the boundary separating the two classes more complex. On the other hand, a low value of $C$ allows more non separable points, and therefore, has a simpler boundary. Only a small fraction of the $\beta_i$ coefficients are nonzero. The corresponding pairs of $x_i$ entries are known as support vectors and they fully define the decision function. Geometrically, the support vectors are the points lying near the separating hyperplane. $K(x_i, x_j) = \phi(x_i).\phi(x_j)$ is called the kernel function. The kernel function may be linear or nonlinear, like polynomial, sigmoidal, radial basis functions (RBF), etc. RBF kernels are of the following form:

$$K(x_i, x_j) = e^{-\gamma |x_i - x_j|^2} \tag{6}$$

where $x_i$ denotes the $ith$ data point and $\gamma$ is the weight. In this paper, the above mentioned RBF kernel is used. In addition, the extended version of the two-class SVM that deals with multiclass classification problem by designing a number of one against all two-class SVMs, is used here.

## 3 Proposed RotaSVM

Consider a training set $\pounds = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of $N$ independent instances, in which each $(x_i, y_i)$ is described by an input attribute vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in \mathbb{R}^d$ and a class label $y_i$. In a classification task, the goal is to use the information only from $\pounds$ 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 $Y$ be a column vector of size $N$, containing the outputs of each training instance in $\pounds$. Moreover, $\pounds$ can also be expressed by concatenating $X$ and $Y$ vertically, that is, $\pounds = [XY]$. Also let $F = \{X_1, X_2, \ldots, X_d\}^T$ be the attribute or feature set composed of $d$ input attributes or features and $\omega$ be the set of class labels $\{\omega_1, \omega_2, \ldots, \omega_c\}$, from which $Y$ takes values.

In RotaSVM, SVM runs $T$ number of times with different rotational feature set. During the training of each SVM, the feature set $F$ is randomly split into $\mathcal{S}$ ($\mathcal{S}$ is an input parameter of RotaSVM) subsets, which may be disjoint or intersecting. To maximize the chance of high diversity, disjoint subsets are chosen. Subsequently, a submatrix $X_{t,s}$, where $t$ is the timestamp of the SVM classifier runs and $s$ is the subset number, is created with the attributes in $F_{t,s}$. From this submatrix, $X_{t,s}$, a new bootstrap sample $X'_{t,s}$ of size 75% is selected. Thereafter, PCA technique is applied to each subset to obtain a matrix $D_{t,s}$ where all principal components are retained in order to preserve the variability information in

the data. Thus, $\mathcal{S}$ axis rotations take place to form the new attributes for SVM classifier. Subsequently, the matrix $D_{t,s}$ is arranged into a block diagonal matrix $R_t$. To construct the training set for classifier $SVM_t$ the rows of $R_t$ are rearranged, so that they correspond to the original attributes in $F$. The rearranged rotation matrix is $R_t^a$ and training set for classifier $SVM_t$ is $[XR_t^a, Y]$. Note that the reason behind selecting 75% is to avoid getting identical coefficients of the principal components if the same attribute subset is chosen for different classifiers and to increase the diversity among the ensemble classifiers. Details of RotaSVM are mentioned in Algorithm 1.

In the testing phase, given a test sample $\mathcal{I}$, let $SVM_{t,i}(\mathcal{I}R_t^a)$ be the posterior probability produced by the classifier SVM$_t$ on the hypothesis that $\mathcal{I}$ belongs to class $\omega_i$. Then the confidence for a class is calculated by the average posterior probability of combination SVMs:

$$\psi_i(\mathcal{I}) = \frac{1}{T}\sum_{t=1}^{T} SVM_{t,i}(\mathcal{I}R_t^a), (i = 1, 2, \ldots, c) \tag{7}$$

Thereafter, $\mathcal{I}$ is assigned to the class with the largest confidence. Note that while running the RotaSVM algorithm to solve a classification task, some parameters like $T$ and $\mathcal{S}$ are needed to be specified in advance.

**Algorithm 1.** RotaSVM

**Require:** For Training
$X$, Data Set
$Y$, Class Label
$T$, Number of time SVM runs
$\mathcal{S}$, Number of Feature Sets
**Require:** For Testing
$\mathcal{I}$, A data object to classify
**Ensure:**
Class lable of $\mathcal{I}$
Prediction Error of RotaSVM

1: **for** $(t = 1, 2, \ldots, T)$ **do**
2: Randomly split the attribute set $F$ into $\mathcal{S}$ subsets, $F_{t,s}$ where $(s = 1, 2, \ldots, \mathcal{S})$.
3: **for** $(s = 1, 2, \ldots, \mathcal{S})$ **do**
4: Create submatrix $X_{t,s}$ using $X$ and $F_{t,s}$.
5: Create a new bootstrap sample $X'_{t,s}$ of size 75% form $X_{t,s}$.
6: Apply PCA on $X'_{t,s}$ to obtain the coefficient matrix $D_{t,s}$.
7: **end for**
8: Arrange the matrices $D_{t,s}(s = 1, 2, \ldots, \mathcal{S})$ into a block diagonal matrix $R_t$.
9: Construct the rotation matrix $R_t^a$ by rearranging the rows of $R_t$.
10: Train the classifier $SVM_t$ using $[XR_t^aY]$ as the training set.
11: **end for**
12: Test the sample $\mathcal{I}$ using different $SVM_t$ and compute average posterior probability to assign class label.
13: **return** ***Class label of*** $\mathcal{I}$ ***and Prediction Error of RotaSVM.***

# 4 Empirical Results

The effectiveness of the RotaSVM is quantitatively measured by comparing it with Bagging, AdaBoost, MultiBoost and Rotation Forest for 10 real-life data sets. In this section, details of data sets, performance metrics and results are discussed.

## 4.1 Data Sets

Table 1 gives the information about data sets, where their different characteristics and variety of fields are described in first three columns by giving the name, sample size and number of classes of each data set, respectively. The last column summarize the information of total number of input attributes in the data sets. All the data sets taken in this experiment, have only numerical attributes. During the pre-processing of data, instances that contain missing values are deleted from the data sets.

**Table 1.** Summery of the data sets

| Data Set | Size | Classes | Total number of input attributes |
|---|---|---|---|
| Balance | 625 | 3 | 4 |
| BCW | 691 | 2 | 9 |
| Dermatology | 366 | 6 | 34 |
| Ecoli | 366 | 8 | 7 |
| Glass | 214 | 6 | 10 |
| Ionosphere | 351 | 2 | 34 |
| Iris | 150 | 3 | 4 |
| Sonar | 208 | 2 | 60 |
| Vehicle | 94 | 4 | 18 |
| Wine | 178 | 3 | 13 |

## 4.2 Experimental Setup

The experimental settings are kept same as used in [11–15] for other ensemble based classifiers such as Bagging, AdaBoost, MultiBoost and Rotation Forest. The classification tree for all these algorithms is created by the "Treefit" algorithm in Matlab. The implementations of these methods are done in Matlab software with version 7.1. For RotaSVM and Rotaion Forest, we have fixed the ensemble size or the number of times SVM and Decision Tree runs arbitrarily at $T = 6$. The value of $\mathcal{S}$ is not fixed for each data set, thus we have adjusted it manually. Here, each method run 20 times and their prediction errors are summarized by computing mean, standard deviation as well as the *kappa* index [28]. Finally, statistical test has been conducted to show the superiority of the results produced by RotaSVM. Note that RBF (Radial Basis Function) kernel is used for SVM in our experiment. Here the parameters of SVM such as $\gamma$ for kernel function and the soft margin $C$ (cost parameter), are set to be 0.5 and 2.0, respectively.

**Table 2.** Mean and standard deviation of prediction errors (expressed in %) for 20 runs of each method on real-life data sets

| Data Set | RotaSVM | Bagging | AdaBoost | MultiBoost | Rotation Forest |
|---|---|---|---|---|---|
| Balance | 8.32 ±0.91 | 15.03 ±0.93● | 21.62 ±0.75● | 18.03 ±0.83● | 9.62 ±0.72● |
| BCW | 3.47 ±0.29 | 3.45 ±0.34 | 3.15 ±0.35 | 3.13 ±0.30 | 2.89 ±0.33○ |
| Dermatology | 2.33 ±0.78 | 3.6 ±0.78● | 3.14 ±0.80● | 3.02 ±0.79● | 2.62 ±0.68 |
| Ecoli | 15.80 ±2.11 | 17.23 ±1.36● | 15.7 ±1.16 | 14.99 ±1.16○ | 16.37 ±1.33● |
| Glass | 1.25 ±0.71 | 24.53 ±0.92● | 24.56 ±0.88● | 24.37 ±0.83● | 24.39 ±0.56● |
| Ionosphere | 1.33 ±0.59 | 8.46 ±0.74● | 6.3 ±0.77● | 6.27 ±0.80● | 5.53 ±0.72● |
| Iris | 2.67 ±0.94 | 4.7 ±1.23● | 5.33 ±1.08● | 5.8 ±1.15● | 4.37 ±1.00● |
| Sonar | 12.98 ±0.68 | 23.08 ±1.82● | 17.98 ±2.14● | 18.63 ±2.23● | 17.26 ±2.24● |
| Vehicle | 11.71 ±1.51 | 25.79 ±0.97● | 23.68 ±1.06● | 23.65 ±0.89● | 21.82 ±1.00● |
| Wine | 1.49 ±0.38 | 4.55 ±1.77● | 3.31 ±0.81● | 3.23 ±1.01● | 5.11 ±1.77● |

"●" indicates that RotaSVM is significantly better and "○" denotes that RotaSVM is significantly worse at the significance level $\alpha = 0.05$.

**Table 3.** One-tailed paired *t*-test results of RotaSVM in comparison with other methods for real-life data sets

| Algorithm | | Bagging | AdaBoost | MultiBoost | Rotation Forest |
|---|---|---|---|---|---|
| | *t*-test Result | | | | |
| RotaSVM | Win | 9 | 8 | 8 | 8 |
| | Tie | 1 | 2 | 1 | 1 |
| | Loss | 0 | 0 | 1 | 1 |

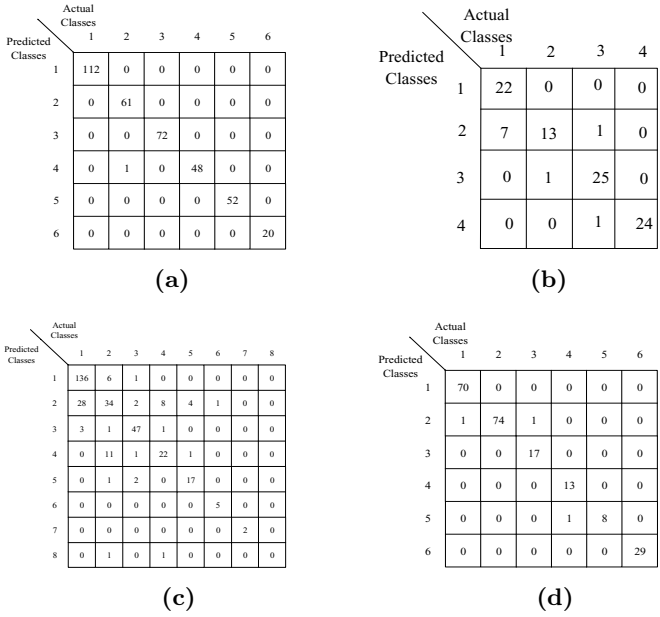


**Fig. 1.** Best Confusion matrix out of 20 runs for (a) Dermatology (b) Vehicle (c) Ecoli and (d) Glass data sets

**Table 4.** Average values of *Kappa* Index for different data sets

| Data Set | RotaSVM | Bagging | AdaBoost | MultiBoost | Rotation Forest |
|---|---|---|---|---|---|
| Balance | 0.87 | 0.65 | 0.59 | 0.62 | 0.71 |
| BCW | 0.91 | 0.91 | 0.92 | 0.92 | 0.93 |
| Dermatology | 0.93 | 0.90 | 0.91 | 0.92 | 0.93 |
| Ecoli | 0.64 | 0.62 | 0.64 | 0.65 | 0.63 |
| Glass | 0.95 | 0.55 | 0.55 | 0.56 | 0.56 |
| Ionosphere | 0.94 | 0.86 | 0.87 | 0.87 | 0.88 |
| Iris | 0.92 | 0.88 | 0.87 | 0.86 | 0.87 |
| Sonar | 0.78 | 0.55 | 0.61 | 0.60 | 0.61 |
| Vehicle | 0.80 | 0.53 | 0.55 | 0.55 | 0.57 |
| Wine | 0.93 | 0.89 | 0.90 | 0.91 | 0.89 |

### 4.3 Results

In Table 2, the mean and standard deviation of the prediction errors (expressed in %) for each method on 10 date sets are reported, where the values of standard deviation are followed after "±". RotaSVM gives consistent results for all data sets. Moreover, the minimum error is achieved by RotaSVM for "Glass" data set. In order to see whether RotaSVM is significantly better or worse than other methods from the statistical viewpoint, a one-tailed paired *t*-test [24] is performed at $\alpha = 0.05$ significance level. The results for which a significant difference of RotaSVM with other methods are found and marked with a bullet or an open circle next to the values of standard deviation in Table 2. A bullet indicates that RotaSVM is significantly better than the other methods and an open circle gives that RotaSVM performs significantly worse than the corresponding method. As it can be seen from Table 3, the "Win-Tie-Loss" information is given, where the "Win" value is the number of data sets on which RotaSVM performs significantly better than the other methods, the "Tie" is the number of data sets on which the differences between the performance of RotaSVM and that of the compared methods are not significantly better, and the "Loss" denotes the number of data sets on which RotaSVM behaves significantly worse than the corresponding algorithm. While compared RotaSVM with Rotation forest, the statistically significant difference is favourable in 8 cases, unfavourable in 1 cases and not significant in 1 cases. Similarly, RotaSVM has been found to outperform Bagging, AdaBoost and MultiBoost in most of the cases.

Here, the confusion matrix is also computed to measure the performance of RotaSVM. The confusion matrix [29] is the result of the classification phase where each classified instance is mutual exclusively located in the matrix. For every cell in the matrix, the column represents the original or actual classes and the row represents the classes as the classification method predicted. The diagonal of the matrix represents the ideal case in which the instances are correctly classified. All the off diagonal cells represent miss classified instances. An important advantage of using the confusion matrix is the ability to consider the performance of the classification method. Fig 1 shows the four best confusion matrices produced by the RotaSVM for Dermatology, Vehicle, Ecoli and Glass data sets, respectively. The accuracy assessment of different methods have also

been justified by measuring *kappa* index, and reported in Table 4. The higher value of *kappa* (close to 1) indicates better accuracy. For most of the cases, it is found from the Table 4 that the *kappa* values are also better for RotaSVM.

## 5 Conclusions

In this paper, RotaSVM ensemble classifier is developed that uses rotational feature selection approach with the integration of Support Vector Machine classifier. To generate the number of predefined outputs of Support Vector Machines, the training data is prepared by splitting the feature set randomly into $\mathcal{S}$ number of subsets. Subsequently, principal component analysis is used for each subset to generate new feature sets, which are reassembled to preserve the variability information in the data. Such features are later used to train Support Vector Machines. Finally, the classification is done by computing average posterior probability. The results, demonstrate the superiority of the RotaSVM quantitatively by comparing it with Bagging, AdaBoost and MultiBoost for 10 real-life data sets taken from UCI Machine Learning Repository. Statistical test like one-tailed paired $t$-test has been performed to show the superiority of the result produced by RotaSVM.

There are still some interesting issues in RotaSVM that are needed to investigate in future. In this regards, trade-off between the parameters like T and S can be archived automatically using multiobjective optimization techniques [30, 31]. Moreover, RotaSVM can be applied for pixel classification of satellite imagery [20, 32, 33], microarray classification [34, 35], protein translational modification site prediction [36, 37], human leukocyte antigen class II binding peptide prediction [38] ect. The authors are currently working in this direction.

**Acknowledgments.** This work is partially supported by Erasmus Mundus Mobility with Asia (EMMA) fellowship grant 2012 from European Union (EU) and University with Potential for Excellence (UPE) - Phase II project grant from University Grants Commission (UGC) in India. Moreover, one of the authors, Mr. I. Saha, would also like to express sincere thanks to the All India Council for Technical Education (AICTE) for providing National Doctoral Fellowship (NDF).

## References

1. Benediktsson, J.A., Kittler, J., Roli, F. (eds.): MCS 2009. LNCS, vol. 5519. Springer, Heidelberg (2009)
2. Haindl, M., Kittler, J., Roli, F. (eds.): MCS 2007. LNCS, vol. 4472. Springer, Heidelberg (2007)
3. Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.): MCS 2005. LNCS, vol. 3541. Springer, Heidelberg (2005)
4. Roli, F., Kittler, J., Windeatt, T. (eds.): MCS 2004. LNCS, vol. 3077. Springer, Heidelberg (2004)

5. Windeatt, T., Roli, F. (eds.): MCS 2003. LNCS, vol. 2709. Springer, Heidelberg (2003)
6. Roli, F., Kittler, J. (eds.): MCS 2002. LNCS, vol. 2364. Springer, Heidelberg (2002)
7. Kittler, J., Roli, F. (eds.): MCS 2001. LNCS, vol. 2096. Springer, Heidelberg (2001)
8. Kittler, J., Roli, F. (eds.): MCS 2000. LNCS, vol. 1857. Springer, Heidelberg (2000)
9. Maulik, U., Bandyopadhyay, S., Saha, I.: Integrating clustering and supervised learning for categorical data analysis. IEEE Transactions on Systems, Man and Cybernetics, Part A 40(4), 664–675 (2010)
10. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Unsupervised and supervised learning approaches together for microarray analysis. Fundamenta Informaticae 106(1), 45–73 (2011)
11. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
12. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 148–156 (1996)
13. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
14. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
15. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation Forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619–1630 (2006)
16. Giacinto, G., Roli, F.: An approach to the automatic design of multiple classifier systems. Pattern Recognition Letters 22(1), 25–33 (2001)
17. Perdisci, R., Ariu, D., Fogla, P., Giacinto, G., Lee, W.: McPAD: A multiple classifier system for accurate payload-based anomaly detection. Computer Networks 53, 864–881 (2009)
18. Giacinto, G., Perdisci, R., Rio, M.D., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion 9(1), 69–82 (2008)
19. Giacinto, G., Roli, F., Didaci, L.: Fusion of multiple classifiers for intrusion detection in computer networks. Pattern Recognition Letters 24(12), 1795–1803 (2003)
20. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: SVMeFC: SVM ensemble fuzzy clustering for satellite image segmentation. IEEE Geoscience and Remote Sensing Letters 9(1), 52–55 (2011)
21. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Advances in Neural Information Processing Systems 7, pp. 231–238 (1995)
22. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152 (1992)
23. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/mlrepository.html`
24. Ferguson, G.A., Takane, Y.: Statistical analysis in psychology and education (2005)
25. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
26. Goh, K.S., Chang, E.Y., Li, B.: Using one-class and two-class SVMs for multiclass image annotation. IEEE Transactions on Knowledge and Data Engineering 17(10), 1333–1346 (2005)

27. Juang, C.F., Chiu, S.H., Shiu, S.J.: Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 37(6), 1077–1087 (2007)
28. Cohen, J.A.: Coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
29. Marom, N.D., Rokach, L., Shmilovici, A.: Using the confusion matrix for improving ensemble classifiers. In: Proceedings of the IEEE Twenty Sixth Convention of Electrical and Electronics Engineers, pp. 555–559 (2010)
30. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
31. Saha, I., Maulik, U., Plewczynski, D.: A new multi-objective technique for differential fuzzy clustering. Applied Soft Computing 11(2), 2765–2776 (2011)
32. Maulik, U., Saha, I.: Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognition 42(9), 2035–2149 (2009)
33. Maulik, U., Saha, I.: Automatic fuzzy clustering using modified differential evolution for image classification. IEEE Transactions on Geoscience and Remote Sensing 48(9), 3503–3510 (2010)
34. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Improvement of new automatic differential fuzzy clustering using svm classifier for microarray analysis. Expert Systems with Applications 38(12), 15122–15133 (2011)
35. Saha, I., Plewczynski, D., Maulik, U., Bandyopadhyay, S.: Improved differential evolution for microarray analysis. International Journal of Data Mining and Bioinformatics 6(1), 86–103 (2012)
36. Plewczynski, D., Basu, S., Saha, I.: AMS 4.0: Consensus prediction of post-translational modifications in protein sequences. Amino Acids 43(2), 573–582 (2012)
37. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Fuzzy clustering of physicochemical and biochemical properties of amino acids. Amino Acids 43(2), 583–594 (2012)
38. Saha, I., Mazzocco, G., Plewczynski, D.: Consensus classification of human leukocyte antigens class II proteins. Immunogenetics 65, 97–105 (2013)