Optimization of Ensemble based Decision using PSO

Asma Kausar, M. Ishtiaq, M. Arfan Jaffar, Anwar M. Mirza

***Abstract*—In matter of great importance, it is the innate behaviour for human beings to seek more than one consultant before making a decision. In such cases we weigh the individual opinions of experts on the basis of their competence and then combine them through some thought process in order to finalize the decision. In this paper we have proposed an idea of Particle Swarm Optimization (PSO) in order to optimize these weights which then better evaluate the competence of an expert. Weighted Majority Voting (WMV) is the most popular technique used to combine such opinions in an ensemble based classification. The weights associated to each base classifier in WMV on the basis of its competence are optimized under the influence of the basic idea of PSO. PSO has shown the stable performance on the selected datasets from UCI Repository and generally improved the performance of an ensemble system.**

***Index Terms*— Classification, ensemble, PSO, Decision Profile, k-fold, WMV, classifier fusion**

I. Introduction

During the pattern analysis we come across various overlapping samples of a dataset. Few of these samples are better analysed using one classifier whereas some other classifier may work better for other type of samples. Instead of making complex solution or a poor decision due to the bad choice of classifier, we can make a risk free decision through an ensemble that may be a little lower in performance.

Due to its simplicity and performance an ensemble based classification has been widely used to improve the confidence of making right decision. The main goal of our ensemble is to improve the confidence of making right decision, by weighing various opinions and combining them through some machine learning techniques to reach a final decision. Classifier fusion technique - all base classifiers are trained over the entire feature space - has been adopted by WMV combiner, which is considered as the most reliable technique of the latest era. Though the diversity - a basic strategy of an ensemble system- is mainly achieved by using different base classifiers like Linear Discremenent Classifier, Quadratic Discremenent Classifier, K-Nearest Neighbour Classifier and Back Propagation. At the end resulting decision is optimized using PSO which is a stochastic population-based computer algorithm modelled on swarm intelligence.

An ensemble based classification has been widely used for many data mining applications. It has provided dramatically better results in case of multi classification applications like "multisensory data fusion" [1], real time object detection, pose estimation (medical imaging) [2], and analysis of fatal desease-Al-Zhemer.

Manuscript received March 9, 2010. This work was supported by the FAST and HEC.
Asma Kausar NUCES, FAST, Islamabad i080857@nu.edu.pk
M. Ishtiaq, NUCES, FAST, Islamabad m.ishtiaq@nu.edu.pk
M. A. Jaffar are with the National University of Computer & Emerging Sciences, FAST, Islamabad (corresponding author: +92 3335375365; e-mail: arfan.jaffar@nu.edu.pk).
Anwar M. Mirza NUCES, FAST, Islamabad, anwar.m.mirza@nu.edu.pk

Ensemble approach has provided better results for the applications where we are concerned with more feature of a dataset or more categories to be targeted. So far we do not have a sufficient theory to explain the performance of an ensemble system compared to single classifiers. Most of the developments in the area are based on the simplification and assumptions and they mostly consider special cases [3], [4], [5]. Due to the difficulty in choosing a suitable combination, recent offspring's of pattern recognition and machine learning, ensemble based classifiers still enjoy many heuristic ideas. Many experimental studies have been published in [6] to provide guidelines in order to prioritise the ideas according to the application in hand.

In the past, pattern recognition focused on designing single classifiers which become much complex in case of non-linear or difficult applications. This paper is about combining the opinions of pattern classifiers from various domains, to make the new opinion more reliable and better than the individual base classifiers in performance. In this paper, the term expert, classifier and hypothesis are used interchangeably where the goal of an expert system is to obtain the optimized decision, by using classifier fusion technique based on their competence. The final expert/ classifier makes the hypothesis about the classification of a given data instance into one of the predefined categories known as supervised classification. The next section describes a brief history of ensembles based classification. In section 3 the major related work is introduced. Section 4 contains the proposed idea of optimization based on PSO which is followed by the experimental setup and results, provided in section 5. Section 6 concludes the comparison of proposed technique with the existing ones followed by future work.

II. History

Though a modern area of pattern recognition and machine learning has started just a decade ago but, in fact, combining classifiers is much older. Although it began with the idea of viewing the classifier output as a new feature vector, traced back to Sbystyen [7] in his book Decision Making Processes in Pattern Recognition in 1962. In his book Sbystyen proposes a cascade machine in which the output of a classifier is fed as an input of the next classifier in a sequence and so on. Though the earliest work on ensemble system is considered to be done by Dasarthy and Sheela's 1979 paper [8]. In this paper, two or more classifier models are used for partitioning the feature space depending on the location of the input. In 1981, Rastrigin and Erinstein introduced dynamic classifier selection [9] in their book [10] which unfortunately reached only to the Standered four-fold cross-validation set-up. The book by "Barabash" [11], was published in 1983, which contain meaningful results about the majority vote for classifier combination. The generalization performance of a neural network was shown to be improved using an ensemble of similarity configured neural networks by Hensen and Salamon [12] while Schapire proved that a strong classifier in probably approximately correct (PAC) sense can be generated

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

by combining the weak classifiers through boosting [13]. The research in ensemble systems has expanded to many creative names and ideas. Few of the examples are classifier systems[8], mixture of experts [14], stacked generalization [15], combination of multiple classifiers [16], [17], [18], dynamic classifier selection [19], classifier fusion [20],[21], committees of neural networks [22], voting pool of classifiers [23], classifier ensembles [24].

The above mentioned approaches usually differ from each other with respect to the specific procedures used for generating base classifiers, and the strategy employed for combining the classifiers. Two main strategies are used for combining the classifier outputs; one is *classifier selection* in which each classifier is trained to become an expert in some local area of feature space. The second is *classifier fusion* in which all classifiers are trained on the entire feature space called weak classifiers whose decisions are then merged to produce a strong single expert. In the later approach, empirical distribution method is applied along with hybrid approaches for the gain of maximum diversity (of base classifiers) and performance(of the final strong classifier). Examples of the above mentioned approach include bagging [25], boosting [26], [27] and many of their variations like Random Forest [28], Pasting Small Votes [29], and Adaboost [30], Learn++.NC [31] etc.

The combination of classifiers can be applied to the classification labels only, or to the specific continuous valued outputs of the individual experts [21], [32], [33]. The classifier outputs are often normalized to the [0,1] interval, and then named as support given by classifier to each class which is based on class-conditional posterior probabilities [21], [34]. Such type of support allows an ensemble to apply algebraic combination techniques like majority voting, maximum, minimum, sum, product or any other combinations of posterior probabilities [9], [33], [35]. Fuzzy integral [20], the Dempster Shafer based classifier fusion [18], [36] and Decision Templates [21], [32], [37] are the most recently used combination techniques for soft label outputs. Theoretical models were developed for combining Discrenement functions in [38], [39] and six commonly used combination rules are compared for their performance to predict posterior probabilities in [35].

III. Related Work

A. Classifier Combination via Continuous-Valued Outputs

The degree of support for a given input can be interpreted in different ways, the two most common ones are

i). Confidences in suggested labels and

ii). Estimates of the posterior probabilities for the classes. Let $x \in R^n$ be a feature vector and $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$ be the set of class labels. Each classifier D_t in the ensemble $D = \{D_1, D_2, \dots, D_T\}$ outputs C degrees of support. Without loss of generality we can assume that all C degrees are in the interval [0, 1], that is, $D_t: R^n \to [0,1]^c$. Denoted by $d_{t,j}(x)$, the support that classifier D_i gives to the hypothesis that the instance x belongs to class ω_j. The larger the support, the more likely the class label ω_j. The C classifier outputs for a

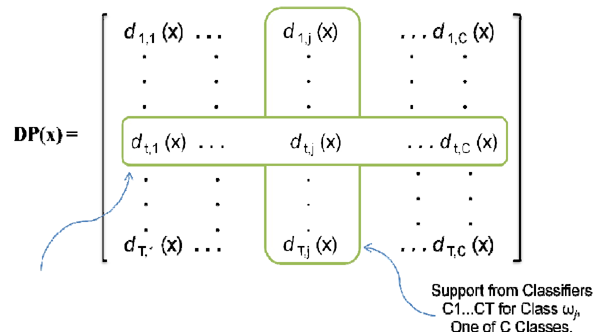


Figure 1 Decision Profile for a given instance x

TABLE I
Summary of Arithmetic Combination Rules

Rule	Formula	Combination function
Mean Rule	$\mu_j(x) = \frac{1}{T}\sum_{t=1}^{T} d_{t,j}(x)$	F = mean
Max Rule	$\mu_j(x) = \frac{1}{T}\max_{t=1\dots T}\{d_{t,j}(x)\}$	F = max
Median Rule	$\mu_j(x) = \frac{1}{T}\mathrm{median}_{t=1\dots T}\{d_{t,j}(x)\}$	F = median
Minimum Rule	$\mu_j(x) = \frac{1}{T}\min_{t=1\dots T}\{d_{t,j}(x)\}$	F = min

particular input x can be organized in a decision profile ($DP(x)$) as a matrix shown in Fig. 1. Kuncheva et al. define the decision profile matrix in [21], which allows the following combination rules from an unified perspective. The decision profile matrix $DP(x)$, for an instance x, consists of elements $d_{t,j}(x) = [0, 1]$, which represent the support given by the t[th] classifier to class ω_j. The rows of $DP(x)$, therefore, represent the support given by individual classifiers to each of the classes, whereas the columns represent the support received by a particular class from all classifiers. Combination methods that use only one column of $DP(x)$ at a time are called "class-conscious combiners" [21] for examples the simple and weighted average, product, and order statistics. Alternatively, we ignore the context of $DP(x)$ and treat the values $d_{t,j}(x)$ as features in a new feature space, named as intermediate feature space. The final decision is made by another classifier that takes the intermediate feature space as input and outputs a class label. This class of methods is named "class-indifferent combiners" according to [21].

All of the class-conscious combiners are idempotent by design. The two main groups of this category are;

- *Non-Trainable Combiners:* also called fixed combiners which do not need to train any extra parameters. In this case the ensemble is ready for the operation as soon as the base classifiers are trained. Examples are algebraic combiners, some of whose formulae are given in table 1.
- *Trainable Combiners:* Some extra parameters are required here like the competence and support in order to prioritize the decision of classifiers. Its examples include Weighted Average which is of three types depending on it calculation of weights [40].

B. Classifier Combination via Label Outputs

Assume that the label outputs of the classifiers are given as C-dimensional binary vectors $[d_{i,1}, \dots, d_{t,j}]^T \in [0,1]^c$, j =

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$[1, \dots, C]$ and $t = [1, \dots, T]$, where $d_{i,j} = 1$ if T_i labels x in ω_j, and 0 otherwise. "Majority Vote" is the main example of this group which became established in 1356 for the election of German kings. The ensemble decision for the plurality voting- a type of majority voting- decides the category/ class that is predicted by atleast one more than half the number of classifiers. It can be described by choosing class ω_j, if

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} d_{t,j} \tag{1}$$

C. Weighted Majority Voting

If the experts in a system do not provide identical accuracy, then it is better to give more power to the more competent expert in making the final decision. let us assume the decision of hypothesis h_t on class ω_j as $d_{t,j}$, such that $d_{t,j}$ is 1, if h_t selects ω_j and 0, otherwise. Further assume that we have a way of estimating the future performance of each classifier, and we assign a weight w_t to classifier h_t in proportion to its estimated performance. According to above assumption, the classifiers whose decisions are combined through weighted majority voting will chose class ω_j, if

$$\sum_{t=1}^{T} w_t d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} w_t d_{t,j} \tag{2}$$

We can normalize these weights to sum up to 1; however normalization does not change the outcome of the WMV criterion. Normally we use the performance of a classifier on a separate validation dataset, or on the training dataset, as an estimate of the future performance of that classifier. This approach is followed by most commonly used ensemble technique, Adaboost. A detailed discussion about WMV can be found in [41].

Assigning the weights to the classifiers is not sufficient to guarantee the maximum performance. The prior probabilities of the classes must also be considered. The optimal weights only magnify the relevance of an individual classifier based on its competence but it does not consider the performance of the other member classifiers of the ensemble team.

D. Optimising confidence of Classifier Decision

Calculating the coefficients (weights) for a classifier on the basis of its competence is the major focus of most commonly used ensemble techniques. In this paper we optimize these weights using Particle Swarm Optimisation (PSO) for making our decision more confident. PSO is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [42]. It is inspired by social behaviour of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA).

PSO is a zero-order; non-calculus-based method which means that it does not need gradients making it suitable to solve discontinuous, multimodal and non-convex problems. It includes some probabilistic features in the motion of particles. The system is initialized with a population of random solutions and it searches for the optima by updating generations. Unlike GA, PSO has no evolution operators such as crossover and mutation. It uses the potential solutions, called particles that fly through the problem space by following the current optimum particles. PSO is easy to implement and there are fewer parameters to adjust than those of the other optimization techniques like GA. It has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas wherever GA can be applied. Each particle of swarm has three features according to [42]:

- *Position* (this is the i^{th} particle at time k, notice vector notation)
- *Velocity* (similar to search direction, used to update the position)
- *Fitness or objective* (determines which particle has the best value in the swarm and also determines the best position of each particle over time.

The basic algorithm of swarm optimization is provided below:

Initial Swarm

- Establish the swarm size (normally 15 to 30)
- Randomly distribute the particles across the design space.

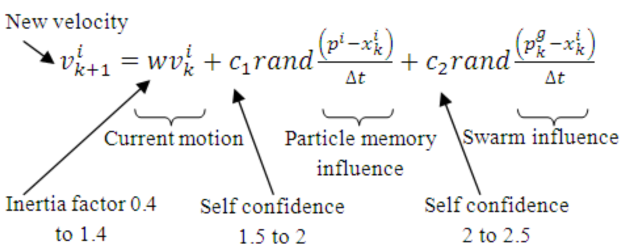
$$x_0^i = x_{min} + rand(x_{max} - x_{min})$$

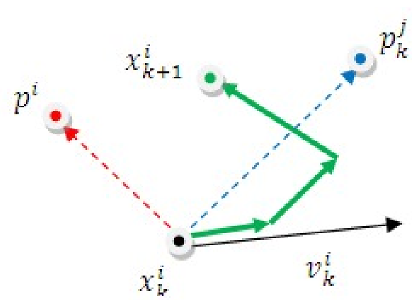
  Where x_{min} and x_{max} are vectors of lower and upper limit values respectively.
- Evaluate the fitness of each particle and store:
  - Particle best ever position(particle memory p^i here is same as x_0^i)
  - Best position in current swarm (influence of swarm p_0^g)
- Generate initial velocity randomly.

$$v_0^i = \frac{x_{min} + rand(x_{max} - x_{min})}{\Delta t} = \frac{\text{position}}{\text{time}}$$

Velocity Update

- To provide search directions it includes deterministic and probabilistic parameters.
- Combine the effect of current motion, particle own memory and swarm influence.





Position Update

- Update the position by velocity vector.

$$x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t$$

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Stopping Criteria

- Maximum change in best fitness smaller than specified tolerance for a specified number of moves (S).

$$\left|f(p_k^g) - f\left(p_{k-q}^g\right)\right| \leq \varepsilon \quad q = 1, 2, \dots, S$$

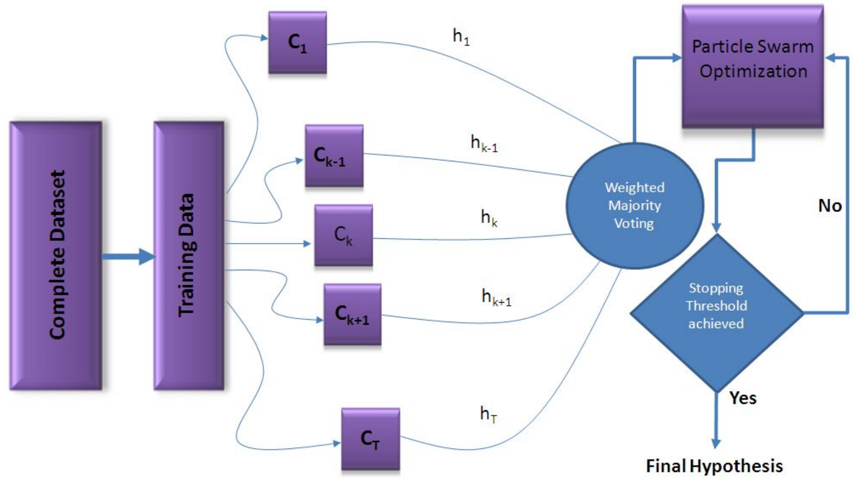


Figure 2 Block diagram of ensemble system optimization

System problems typically include continuous, integer, and discrete design variables. Basic PSO works with continuous variables. There are several methods which allow PSO to handle discrete variables. The simple method of rounding particle position coordinates to the nearest integers provides the best computational performance.

We have trained T number of different type base classifiers $C_1, C_2, \dots, C_T$ on the whole training dataset in order to make a team of hypothesis h_t where $t = 1, 2, \dots, T$. The outputs of these classifiers are fused to attain the stronger classifier using weighted majority voting. The final decision of this heterogeneous ensemble of classifiers is then optimized by using the idea of particle swarm optimization. The process of optimization continues till a predefined threshold is achieved. A block diagram of our proposed method is shown in Fig.2.

IV. Experimental Setup and Results

We have applied different classification techniques on the four multiclass applications from UCI Repository i.e. Heart, Diabetes, Iris and Transfusion. PSO parameters have been fixed exactly according to the basic strategy as proposed in [42]. Four base classifiers-LDA, QDA, KNN, and BP- have been used in our experiments considering their weak reliability, in order to diversify the members. The final results are then compared with these individual classifiers as well as their combinations using fixed algebraic combiners and plurality voting. Average error rates of these combiners are provided in table III.

Practical Swarm Optimization has generously improved the performance of ensemble system on every database. Performance evaluation between simple ensemble and PSO optimized ensemble is shown using the line graph separately on each dataset. A bar chart provides the overview of best performance that can be achieved by individual classifiers and ensemble systems. The proposed optimization has shown the best performance results in seven runs for each dataset except the Transfusion Dataset where it gains the highest performance in nine runs. Fig.7 shows the final comparison of best performance by Weighted Majority Voting combiner with and without optimization.

V. Conclusion

An ensemble based system is more reliable than individual classifiers when we come across the multi classification of nonlinear and complex datasets. The performance of ensemble based system can further be improved by using Practical Swarm Optimization. The idea proposed in this paper is a very simple technique and can also be applied to many other combination techniques of an ensemble. In future we aim to further improve this system by adding on the diversity measures.

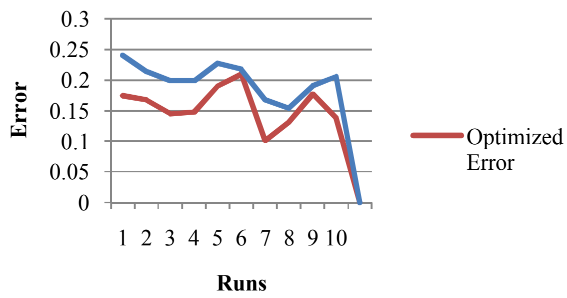


Figure 3 Performance of Simple and Optimised Weighted Majority Voting on Heart Dataset

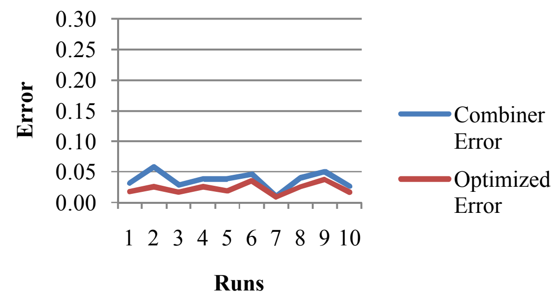


Figure 5 Performance of Simple and Optimised Weighted Majority Voting on Iris Dataset

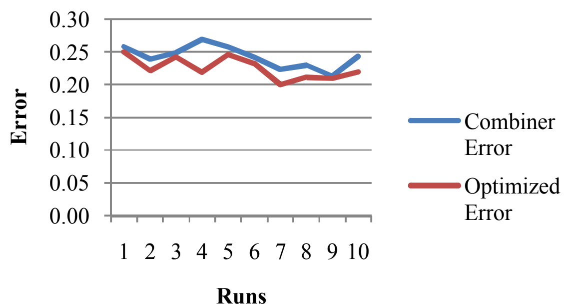


Figure 4 Performance of Simple and Optimised Weighted Majority Voting on Diabetes Dataset

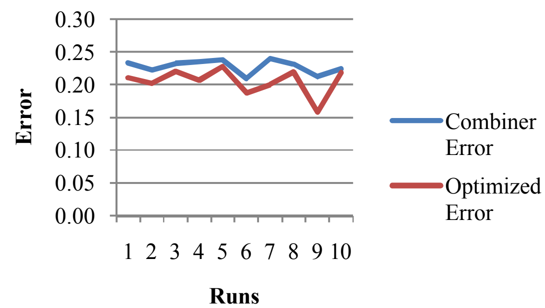


Figure 6 Performance of Simple and Optimised Weighted Majority Voting on Transfusion Dataset.

TABLE III
AVERAGE PERFORMANCE OF DIFFERENT CLASSIFIERS ON DIFFERENT DATASETS

Datasets	Individual Classifier Error				Weighted Combiner Error	Combiners Error						PS Optimized Error
	LDC	QDC	kNNC	BPNC		Prod	Mean	Med	Max	Min	Vote	
Heart	0.173	0.234	0.3566	0.231	0.201	0.197	0.227	0.227	0.227	0.227	0.195	**0.158**
Diabetes	0.241	0.261	0.2929	0.263	0.242	0.243	0.255	0.255	0.255	0.249	0.244	**0.225**
Iris	0.031	0.047	0.0495	0.066	0.036	0.031	0.050	0.054	0.054	0.042	0.037	**0.022**
Transfusion	0.232	0.232	0.2712	0.221	0.227	0.231	0.231	0.231	0.231	0.232	0.230	**0.205**

TABLE IV
BEST PERFORMANCE OF DIFFERENT CLASSIFIERS ON DIFFERENT DATASETS

Data set	Best Individual Classifier				Best Wvote	Best Combiner						PS Optimized
	LDC	QDC	kNNC	BPNC		Prod	Mean	Med	Max	Min	Vote	
Heart	0.142	0.201	0.312	0.179	0.154	0.169	0.185	0.185	0.185	0.201	0.153	**0.1010**
Diabetes	0.216	0.236	0.275	0.233	0.213	0.214	0.223	0.223	0.223	0.225	0.221	**0.2000**
Iris	0.019	0.009	0.038	0.038	0.009	0.019	0.038	0.038	0.028	0.009	0.019	**0.0090**
Transfusion	0.226	0.222	0.241	0.199	0.209	0.224	0.224	0.224	0.222	0.218	0.226	**0.1576**

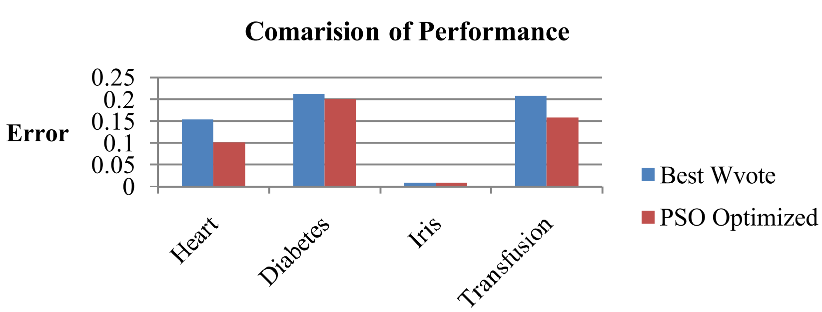


Figure 7 Best Performance of Different Classifiers on Different Datasets

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

ACKNOWLEDGMENT

The authors are thankful to the FAST-National University of Computer & Emerging Sciences and Higher Education Commission of Pakistan for providing funding for the research.

REFERENCES

[1] Devi Parikh and Robi Poliker, "An Ensemble-Based Incremental Learning Approach to Data Fusion," IEEE Transactions on Systems, Man and Cybernetics, Vol. 37, No. 2, April 2007.

[2] J. Zhang, S.K. Zhou, L. McMillan, D. Comaniciu: Joint Real-Time Object Detection and Pose Estimation Using Probabilistic Boosting Network, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, 2007.

[3] G. Fumera and F. Roli, "Performance analysis and comparison of linear combiners for classifier fusion," presented at the 16th Int. Conf. Pattern Recognition, 2002.

[4] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," IEEE Trans. Pattern Anal. Machine Intell., vol. 20, pp. 226–239, Mar. 1998.

[5] ---, "A theoretical study on expert fusion strategies," IEEE Trans. Pattern Anal. Machine Intell., vol. 24, pp. 281–286, Feb. 2002.

[6] Proc. 3rd Int. Workshop Multiple Classifier Systems, vol. 2364, Lecture Notes In Computer Science, F. Roli and J. Kittler, Eds., Cagliari, Italy, 2002.

[7] G. S. Sebestyen. Decision-Making Process in Pattern Recognition. The Macmillan Company, N.Y., 1962.

[8] B.V. Dasarathy and B.V. Sheela, "Composite classifier system design: Concepts and methodology," Proceedings of the IEEE, vol. 67, no. 5, pp. 708–713, 1979.

[9] L. A. Rastrigin and R. H. Erenstein. Method of Collective Recognition. Energoizdat, Moscow, 1981.

[10] K. Woods, W. P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:405–410, 1997.

[11] Y. L. Barabash. Collective Statistical Decisions in Recognition. Radio i Sviaz', Moscow, 1983.

[12] L.K. Hansen and P. Salamon, "Neural network ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993–1001, 1990.

[13] R.E. Schapire, "The strength of weak learnability," Machine Learning, vol. 5, no. 2, pp. 197–227, 1990.

[14] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," Neural Computation, vol. 3, pp. 79–87, 1991.

[15] D.H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.

[16] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," IEEE Transactions on Systems, Man and Cybernetics, vol. 22, no. 3, pp. 418–435, 1992.

[17] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision combination in multiple classifier systems," IEEE Trans. on Pattern Analy. Machine Intel., vol. 16, no. 1, pp. 66–75, 1994.

[18] G. Rogova, "Combining the results of several neural network classifiers," Neural Networks, vol. 7, no. 5, pp. 777–781, 1994.

[19] K. Woods, W.P.J. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 405–410, 1997.

[20] S.B. Cho and J.H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 2, pp. 380–384, 1995.

[21] L.I. Kuncheva, J.C. Bezdek, and R. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," Pattern Recognition, vol. 34, no. 2, pp. 299–314, 2001.

[22] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensemble methods," Neural Computation, vol. 6, no. 6, pp. 1289–1301, 1994.

[23] R. Battiti and A.M. Colla, "Democracy in neural nets: Voting schemes for classification," Neural Networks, vol. 7, no. 4, pp. 691–707, 1994.

[24] L.I. Kuncheva, "Classifier ensembles for changing environments," 5th Int. Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, F. Roli, J. Kittler, and T. Windeatt, Eds., vol. 3077, pp. 1–15, 2004.

[25] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[26] F.M. Alkoot and J. Kittler, "Experimental evaluation of expert fusion strategies," Pattern Recognition Letters, vol. 20, no. 11–13, pp. 1361–1369, Nov. 1999.

[27] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[28] L. Breiman, "Pasting small votes for classifcation in large databases and on-line," Machine Learning, vol. 36, pp. 85–103, 1999.

[29] Y. Freund and R.E. Schapire, "Decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.

[30] Learn++.NC

[31] F.M. Alkoot and J. Kittler, "Experimental evaluation of expert fusion strategies," Pattern Recognition Letters, vol. 20, no. 11–13, pp. 1361–1369, Nov. 1999.

[32] J. Kittler, M. Hatef, R.P.W. Duin, and J. Mates, "On combining classifiers," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226–239, 1998.

[33] M. Muhlbaier, A. Topalis, and R. Polikar, "Ensemble confidence estimates posterior probability," 6th Int. Workshop on Multiple Classifier Sys., in Lecture Notes in Comp. Science, N. Oza, R. Polikar, J. Kittler, and F. Roli, Eds., vol. 3541, pp. 326–335, 2005.

[34] L.I. Kuncheva, "A theoretical study on six classifier fusion strategies," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 281–286, 2002.

[35] Y. Lu, "Knowledge integration in a multiple classifier system," Applied Intelligence, vol. 6, no. 2, pp. 75–86, 1996.

[36] L.I. Kuncheva, "Using measures of similarity and inclusion for multiple classifier fusion by decision templates," Fuzzy Sets and Systems, vol. 122, no. 3, pp. 401–407, 2001.

[37] G. Fumera and F. Roli, "A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp. 942–956, 2005.

[38] Various Authors, "Proceedings of International Workshop on Multiple Classifier Systems (2000–2005)," F. Roli, J. Kittler, T. Windeatt, N. C. Oza, and R. Polikar, Eds. Berlin, Germany: Springer, 2005.

[39] R. A. Jacobs. Methods for combining experts' probability assessments. Neural Computation, 7:867–888, 1995.

[40] C. Diamantini and A. Spalvieri. Quantizing for minimum average misclassification risk. IEEE Transactions on Neural Networks, 9(1):174–182, 1998.

[41] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization," Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia 1995, pp. 1942-1945.

[42] Kennedy, J. and Eberhart, R., Swarm Intelligence, Academic Press, 1st ed., San Diego, CA, 2001.

[43] Venter, G. and Sobieski, J., "Particle Swarm Optimization," AIAA 2002-1235, 43rd AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO., April 2002.