Intelligent combination of Kernels information for improved classification

Abdul Majid, Asifullah Khan and Anwar M. Mirza
Faculty of Computer Science & Engineering, GIK Institute, Ghulam Ishaq Khan (GIK) Institute of Engineering Science & Technology, Topi-23460, Swabi, PAKISTAN
{ majid, akhan and mirza}@giki.edu.pk

Abstract

In this paper, we are proposing a combination scheme of kernels information of Support Vector Machines (SVMs) for improved classification task using Genetic Programming. In the scheme, first, the predicted information is extracted by SVM through the learning of different kernel functions. GP is then used to develop an Optimal Composite Classifier (OCC) having better performance than individual SVM classifiers. The experimental results demonstrate that OCC is more effective, generalized and robust. Specifically, it attains high margin of improvement at small features. Another side advantage of our GP based intelligent combination scheme is that it automatically incorporates the issues of optimal kernel and model selection to achieve a higher performance prediction model.

1. Introduction

In machine learning, there is a considerable interest in getting useful information from a large volume of data. In many engineering and medical applications, there is a strong need to capture data from the environment and classify it accurately in one class or the other. Scientists are also turning to computers to find automatic methods to make sense from data. Bioinformatics has arisen from the need of computer scientists and biologists. This prompted research on new fields of Machine Learning, Data Mining and Knowledge Discovery to develop an intelligent information extraction model [6]-[10]. In order to fulfill the demands of these emerging fields, researchers are always in search of high performance classification model. An improvement in the classification model may influence the overall quality of the system [1].

The main objective of a classification model is to achieve good generalization performance. In disease diagnosis system, practitioners are interested in high performance prediction model. For this purpose, they require an improved Receiver Operating Characteristics (ROC) curve model. Such prediction model is an inevitable solution, which can save thousands of human lives [3].

SVM based discriminant approach is preferred in high dimension image space. SVM models depend on different types of kernel functions. Currently, optimization of SVM models is an active area of research. In order to optimize SVM models, two issues are considered: the selection of suitable kernel function and its associated parameters. So far, no intelligent method has been developed to optimize SVM models [15]-[16]. Most recent work in combining SVM classifiers is presented in [5]. They have used the concept of linear combination of kernels to perform functional (matrix) combination of kernels. This approach uses class conditional probabilities and nearest neighbor techniques for classification.

Now a day, the integration of multiple classifiers has attained a considerable attention for higher classification performance, where the prediction accuracy can be improved by parameter tuning. Such a system is expected to be more accurate than a single classifier [4]. In the integration of classifiers, the deficiency in one classifier can be replaced by the advantage of other. But, due to the lack of general classifiers combination rules, there are many challenges to combine the results provided by each classifier. Mostly, problem specific solutions have been offered. For example, in a simple selective voting scheme of m classifiers requires 2^m combinations; such exhaustive experiments may not be easy in some cases. Various search strategies such as simulated annealing, genetic algorithms and Tabu search may be used to find suitable multiple voting schemes, but such techniques address only one way of finding a suitable combination of classifiers. Since the search space may be complex and there is no guarantee of improvement, the possible combination functions may be very large. Therefore, in order to tackle this complicated problem an intelligent search technique such as Genetic Programming is required. GP has the flexibility to develop an optimal numerical classifier as well as there are good chances for GP based combination of classifiers to perform better.

Previously, GP has been used successfully in combining different classification models such as

artificial neural network, decision trees and linear classifiers to produce a composite classifier [6]-[10]. We are using GP for combining SVM models trained using different kernel functions because different classification models may give us suitable diversity in combination [17]. We address two issues related to the improvement in SVMs models through the following contributions:

Firstly, the genetic combination of the individually trained SVM classifiers enables us to construct an optimal decision space using the decision space of individual kernels. Different kernel functions in the combination may explore the feature space efficiently and a combination has a better chance to exploit the feature space. Secondly, another side advantage, in the combination of SVMs through GP, is that it automatically fulfils the desire of finding optimal model selection for SVMs. This is because GP mechanism incorporates the random constants in addition to variables terminals.

The remaining paper is organized as follows: in section 2, we describe the proposed methodology of combining SVM classifiers and the basic architecture of our proposed classification system. Implementation details are given in section 3. Results and discussion are presented in section 4. Finally, conclusions are given in section 5.

2. Proposed methodology

Our current work is an extension of our previous work. Earlier, the performance of nearest neighbor classifiers [9] and statistical classifiers [10] has been optimized . In the current work, we are extending the idea of heterogeneous combination of classifiers to combine SVM classifiers. The combination of SVM classifiers is carried out by stacking the predictions of individual classifiers. Suppose, there are m kernel functions $K_1, \cdots, K_m$, which are SVM learning algorithms. In order to avoid over-fitting, the dataset S is partitioned into three independent sets by using holdout method, such that $S = \{ X_1 \bigcup X_2 \bigcup X_3 \}$ with examples $s_i = (x_i, y_i)$. A set of individual SVM classifiers $C_1, \cdots, C_m$ are constructed by training different kernel functions on training dataset X_1 such that $C_j = K_j (x_{1i}),\ x_{1i} \in X_1,\ j = 1, 2, \cdots, m \quad i = 1, 2, \cdots, n$. A new features set $(\hat{y}_i^1, \cdots, \hat{y}_i^m)$ is achieved by stacking the predictions of SVM classifiers by using testing dataset X_2 as: $\hat{y}_i^j = C_j (x_{2i}),$ where $x_{2i} \in X_2, \forall\ j = 1, 2, \cdots, m$. (Dataset X_2 is also used to evaluate the performance of evolved OCC and dataset X_3 is used in the testing of classification models.) Now, GP meta-learning process is based on a new training data space of $(\hat{y}_i^1, \cdots, \hat{y}_i^m)$, $\forall\ i = 1, \cdots, n$. The predictions of SVM models are stored in different arrays (L, P and R) and used as unary functions in GP tree as shown in figure 2. GP optimally combine the predictions of individual classifiers to obtain an optimal numerical classifier.

Main modules of our scheme are shown in Figure 1, with double dashed boxes. A brief introduction of each module is as follows:

Formation and normalization of face database: Gender classification problem is taken as a test case. Various databases are combined to form a generalized unbiased database for gender classification task. Different face images are collected from the standard databases ORL, YALE and CVL. In the normalization stage, CSU Face Identification Evaluation System [13] is used to convert all images in the uniform state.

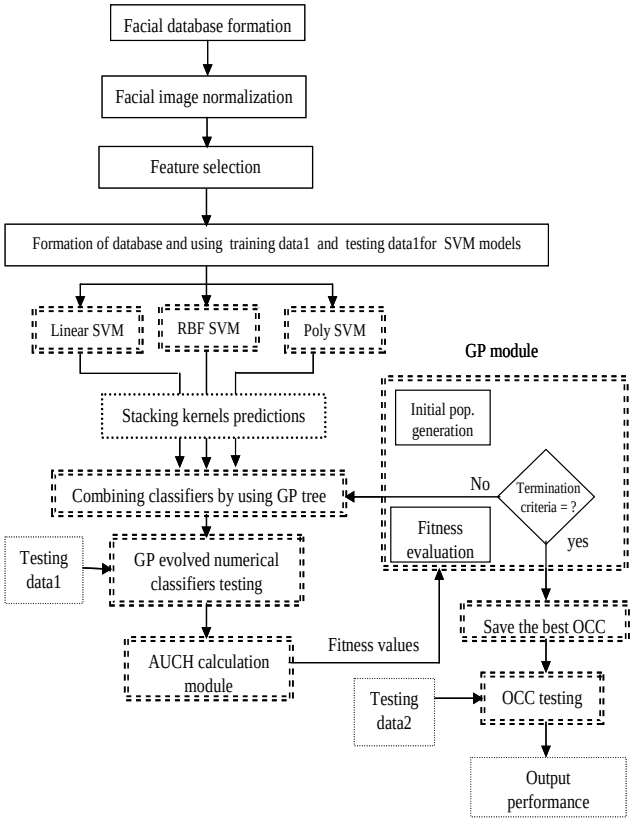

Figure 1 Proposed classification system

Features selection: Feature selection is the task of reducing the dimension by selecting a small set of the features. We are using Iterative Search Margin Based Algorithm (Simba) [12].

Classifier's performance evaluation: In GP evolution process, the choice of fitness function is very important. This function strongly affects the GP programs evolved. In order to obtain ROC curve, the detailed information

can be found in our previous papers [9]-[11]. Area Under the Convex hull (AUCH) of a classifier's ROC is the "maximum realizable" ROC taken, as a measure of the performance in classification models. These parameters are considered an important tools to analyze the performance of classifiers over a wide range of decision thresholds.

SVM Classifiers: SVM performs classification between two classes that has maximum distance to the closest points in the training set [2]. For a linearly separable data, a hyperplane is determined by maximizing the distance between the support vectors. For n data point (x_i, y_i), where $i = 1, \cdots, n$ $x_i \in R^N$ and $y_i \in \{1, -1\}$. the following kernel functions are used and defined as:

$k(x_i, x_j) = x_i^T x_j$ (Linear kernel)

$k(x_i, x_j) = [\gamma(x_i, x_j) + r]^d$, $\gamma > 0$ (Polynomials kernel)

$k(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2)$, $\gamma > 0$ (RBF kernel),

where γ, r, d are the kernel parameters.

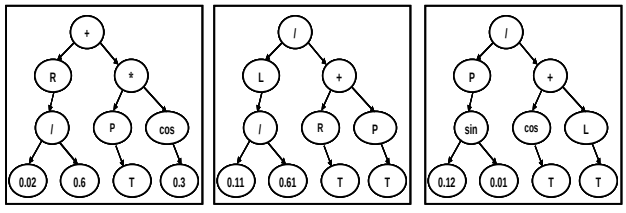

Figure 2 Different combinations in GP trees

2.1. GP module

We represent a classifier as a candidate solution in a tree like-data structure. A random population of classifiers is created in a GP solution space. Next, score each classifier on a classification task, such as measuring accuracy on a set of labeled examples. With the help of GP evolution, each new generation has a slightly higher score. Solution space is refined and converges to the optimal/near optimal solution.

To develop OCC, GPLAB [14] is used. All the necessary settings are given in table 1. We have defined suitable functions, terminals, and fitness criteria. Different parts within the GP module are given as:

GP function set: GP Function set is a collection of various mathematical functions available in the GP module. In GP simulations, we have used simple functions, including four binary floating arithmetic operators (+, -, *, and /), *sin* and *cos*.

GP terminals: To create population of candidate OCC, we consider OCC as a class mapping function which consists of the independent variables and constants. Threshold T is taken as a variable terminal. Randomly generated numbers in the range of [-1, 1] are used as constant terminals in GP tree. In this way, GP is allowed to combine the decision space of different SVM classifiers.

Population initiaization method: Initial popoulation is generated by using ramped half and half method. In this methood, an equal number of individuals are initialized for each depth, with the number of depths considered from two to the initial tree depth value.

Table 1: GP Parameters Settings

Objective:	**To evolve a optimum combined classifier with maximum AUCH**
Function :	*+, -, *, /, gt,le, log, abs, sin and cos*
Special Function:	SVM prediction (*L, P, R*) are used as unary functions
Terminal :	Threshhold T & random constants in the range of [0 - 1]
Fitness :	AUCH of ROC curve.
Expected offspring	rank85
Selection:	Generational
Wrapper:	Positive if >= 0, else Negative.
Pop. Size:	300
Initial population:	Ramped half and half
Operator prob.	Variable
Sampling	Tournament
Survival mechanism	Keep best
Termination:	Generation 80

Fitness evaluation function: In order to assess the performance of individuals in GP population, we have used AUCH of ROC curve as a fitness function. A fitness function grades each individual in the population. It provides feedback to the GP module about the fitness of individuals. Figure 1 shows the usage of fitness function as a feedback. Every individual in the population is evaluated in terms of AUCH of ROC curve.

GP operators: We have used replication, mutation and crossover operators to produce a new GP generation. In mutation a small part of an individual's genome is changed. This small random change often brings diversity in the solution space. Crossover creates an offspring by exchanging genetic material between two individual parents.

3. Implementation details

MATLAB 6.5 environment is used for experimental studies. Using different values of threshold in Simba algorithm [12] for a weight vector *W*, different feature are selected. Each feature set is scaled to the range [-1, +1]. For gender classification, 300 male and 300 female images have been used. The problem of over-fitting is handled in the training of individual SVM classifiers and in the design of OCC, by choosing an appropriate size of training and testing data, carefully setting the parameters in GP simulation.

In holdout method, the dataset *S* is divided into three equal and non-overlapping parts called training data, testing data1 and testing data2. Each part of the data set contains 100 male and 100 female images. Training data is used to train SVM classifiers and testing data1 is used to compute their predictions. These predicted values are scaled in the range of [0, 1]. In order to calculate AUCH of a classifier at the specified feature set using testing data2, first, the decision values of the classifier are obtained. Then, TPR and FPR of the entire test samples in the testing data2 are computed at varying threshold T in the range of [0, 1]. These values of TPR and FPR are plotted to produce ROC curve and then AUCH of ROC curve is determined.

In order to increase the statistical significance of results, the performance of each SVM classifier and OCC is evaluated 10 times by randomly permuting the testing data2. Their average results are, then reported.

4. Results and discussion

Comparison of OCC: In order to explore the optimality of OCC, we have evaluated and compared the performance of OCC and SVM classifiers at different feature sets. AUCHs of ROC curves are obtained at different feature sets as shown in the figure 3 (a) and 3 (b). It is observed that with the provision of more information, TPR increases, while FPR decreases in all classifiers. OCC has outperformed its constituent's kernels with high TPR and low FPR values. Thus, OCC provides more optimal decision space. This type of situation is highly desirable in medical applications, where cost of FPR is very important. For example, before cancer therapy, misclassification of malignant/benign tissues of a weak patient is going to be very costly [11].

Bar chart in Figure 4 shows the comparison of experimental results in a summarized form with various feature sets. It is observed that linear SVM has the lowest value of AUCH at all feature sets. This is due to the nonlinearity in the gender features and linear SVM is unable to resolve it. While, the performances of polySVM and rbfSVM classifier are relatively equal. They are capable of constructing a discriminant nonlinear decision boundary. As far as the performance of OCC is concerned, it is excellent. During GP evolution, OCC has extracted useful information from its constituent kernels decision space. Another side advantage gained is that OCC has its higher performance, specifically at small feature sets of 5, 10 and 20. General order of performance of classifiers in term of AUCH is:

$$OCC > (polySVM \cong rbfSVM) > linearSVM$$

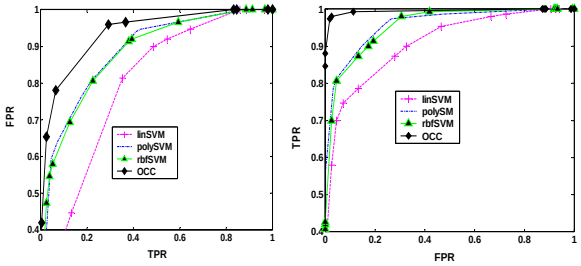

Figure 3 (a) & (b) ROC for 10 & 100 features respectively (*For simplicity, in this figure, we have shown only those points which recline on the convex hull of the ROC curve.*).

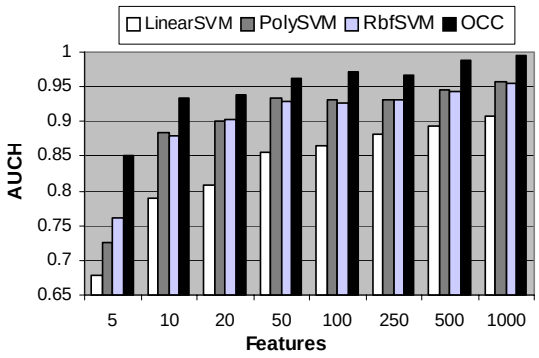

Figure 4 AUCHs of ROC curves of classifiers

Overall performance measure AUCH of AUCH: It is observed in figure 4 that AUCH of classifiers is enhanced with increase in the size of a feature set, so we are proposing a new measure of AUCH of AUCHs, to represent the performance of a classifier in more compact and summarized form. Further, this measure also depicts the robustness of a classifier with respect to the variation in feature sets. In this approach, first, different AUCH values at different feature sets of a classifier are obtained. Then, a graph is plotted between AUCHs versus sizes of different feature sets as shown in figure 5. AUCH of these AUCH curves is also computed in order to find AUCH of AUCHs. Average AUCH of each classifier is also calculated. The difference between AUCH of AUCH and average AUCH of each classifier is determined. The value of difference represents the variation in classifier's performance with respect to the size of feature set. The higher value of difference indicates the lower robustness of a classifier. Bar chart in the figure 6 shows the overall

performance of classifiers in term of AUCH of AUCHs, average AUCH and their percentile difference. It is observed that linear SVM has the lowest AUCH of AUCHs value and the largest percentile difference. However, polySVM and rbfSVM classifiers have comparatively the same AUCH of AUCH values and relatively small percentile difference. Whereas, OCC has the largest AUCH of AUCHs value (0.988) and the smallest percentile difference. These results illustrate the higher optimality and robustness of OCC against the variation in feature sets as compared to its component classifiers. These are the two main advantages that we have achieved in the performance of OCC.

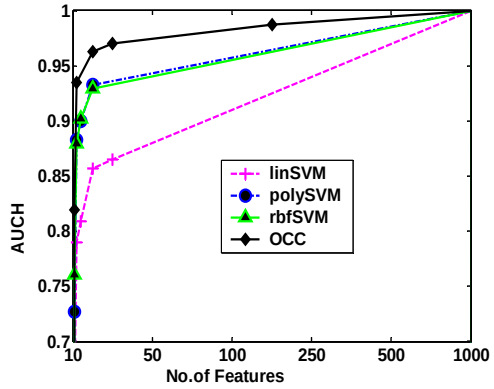

Figure 5 AUCH curves of different classifiers

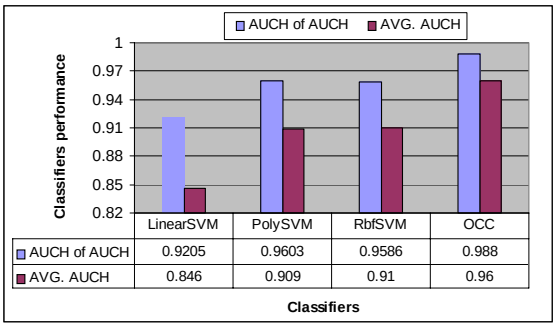

Figure 6 Overall Classifiers performance

OCC performance at feature sets of different sizes: To study the behavior of OCC trained at a particular feature space and to test it at a partially different feature space. The partially different space contains the additional space (with more features) as well as the actual training space. For this purpose, we have constructed different feature sets of 10, 50, 100, 500 and 1000. In table 2, first column shows that OCC-10 is trained at a feature space in which each feature set is of size 10 and tested it at 10, 50, 100, 500 and 1000 features. AUCH of OCC is maximum (0.9344) at a features space of 10 features as compared to other partially different spaces. The table 2 shows the optimal values of AUCH along the diagonal path and normal values at partially different space. It demonstrates that in all columns from top to bottom, there is a gradual increase in the values of AUCH. This is the behavior of OCC as a normal classifier. That is, with the provision of more useful information results in higher output performance. But if we move horizontally, then, there are random AUCH values for OCC. This explains the GP random behavior. Due to its diverse nature, each time, when GP run is carried out to find an optimal solution in a search space that solution may be partially/entirely different from the solution offered by the previous solution space. This proves the optimal behavior of OCC tuned at specific feature space. Therefore, through GP simulation, OCC can be trained at any data space for higher classification task.

Accuracy versus complexity: Figure 7 shows the accuracy versus complexity of the best-evolved OCC at 1000 features upto 60 generations. It is observed that generation-by-generation, there is an improvement in fitness of the best-evolved individual. This improvement is achieved at the cost of complexity in a GP tree, which represents a numerical classifier. During GP crossover and mutation operation, more and more constructive blocks builds up within the numerical evolved classifier. Bloating phenomenon, during GP evolution process minimizes the destruction of these blocks. As a result, average size of individuals increases in each generation. In this way, genome's total number of nodes of the best individual increases and its average tree depth also become very large.

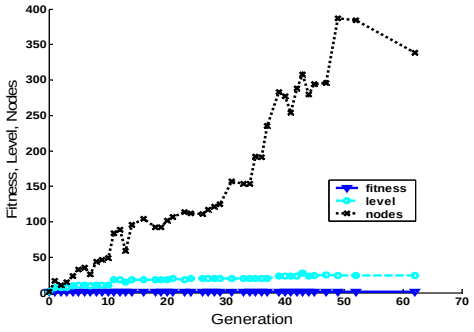

Figure 7 Accuracy vs. complexity graph

Table 2 OCC performances at various feature spaces

Feature size	OCC -10	OCC-50	OCC-100	OCC-500	OCC-1000
FS = 10	**0.934**	0.890	0.863	0.877	0.799
FS = 50	0.900	**0.962**	0.911	0.903	0.843
FS = 100	0.913	0.938	**0.970**	0.931	0.893

FS = 500	0.937	0.941	0.948	**0.987**	0.908
FS=1000	0.945	0.958	0.947	0.953	**0.994**

5. Conclusions

Our GP based technique of developing composite classifier, extracts useful information from its constituent classifiers to improve the classification task. This gain in the performance of OCC is achieved through the genetic combination of SVM classifier without resorting to the manual search of suitable kernel functions and its model selection. From experimental results, it is concluded that OCC is more optimal, robust and generalized at almost all feature sets. During GP evolution process, OCC learns the most favorable distribution within the data space. Using proposed scheme, OCC can be tuned at any binary classification problem, specifically for medical data. Our investigations have explored the GP potential to optimally combine the decision information from its constituent classifiers.

In future, we intend to implement this method for binary/multi classification problems of medical data.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, "*Pattern Classification*," John Wiley & Son s, Inc., New York, 2nd edition (2001),

[2] V. Vapnik, "*Statistical Learning Theory*," New York: John Wiley & Sons Inc (1998) .

[3] K. Rajpoot and N. Rajpoot, "SVM Optimization for Hyperspectral Colon Tissue Cell Classification," *LNCS3217, Springer-Verlag*, (2004), pp. 829-837.

[4] J. Kittler and F. Roli, "Multiple Classifier Systems," Proc. of 2nd International Workshop, *MCS2001*, Cambridge, UK, Lecture Notes in *Computer Science*, vol. 2096, *Springer-Verlag*, (2001).

[5] J. M. Moguerza1, A. Muñoz and I. M. D. Diego, "Improving Support Vector Classification via the Combination of Multiple Sources of Information," *Multiple Classifier Systems I, Springer-Verlag*, (2004), vol. 3138.

[6] W. B. Langdon and S. J. Barrett, "Genetic Programming in Data Mining for Drug Discovery," *in Evolutionary Computing in Data Mining, Physica* Verlag, (2004) ,page 211-235.

[7] B.F. Buxton, W.B. Langdon and S.J. Barrett, "Data Fusion by Intelligent Classifier Combination," *Measurement and Control*, vol. 34, No. 8,(2001), p229-234.

[8] W.B. Langdon and B.F. Buxton, "Genetic programming for combining classifiers," in *GECCO2001*, (2001).

[9] A. Majid, A. Khan, and A.M. Mirza, "Improving Performance of Nearest Neighborhood Classifier Using Genetic Programming," *International conference on machine learning and its application,s ICMLA'04*, Louisville, KY, USA, (2004).

[10] A. Khan, A. Majid, and A.M. Mirza, "Combination and Optimization of Classifiers in Gender Classification Using Genetic Programming," *International Journal of Knowledge-Based Intelligent Engineering Systems*,(2004),Vol. 8, page 1-11.

[11] A. Majid, "Optimization and Combination of Classifiers using Genetic Programming", PhD thesis, *GIK institute of Engineering Science and Technology*, 2005.

[12] R.J. Bachrach, A. Navot, and N. Tishby, "Margin based feature selection - theory and algorithms," proc. of *the 21'st international conference on Machine Learning (ICML)*, (2004).

[13] R. Beveridge, "Evaluation of face recognition algorithms," web site.http:\\cs.colostate.edu\evalfacerec.

[14] http://gplab.sourceforge.net

[15] Y. Guermeur, M. Maumy and F. Sur, "Model selection for multi-class SVMs," ASMDA'05, Brest, (2005).

[16] C. Staelin, "Parameter selection for support vector machines," *Technical report, HP Labs*, Israel, (2002).

[17] G. Brown, J. Wyatt, R. Harris and X. Yao, "Diversity creation methods: A survey and categorization," *Information Fusion*, (2005), 6(1), 5–20.